SIEMENS

CMS X-Tools

User Manual - 01 - Introduction

English

Release 2014-11

Preface	**1**
Scope of Delivery	**2**
Product Characteristics	**3**
Installation	**4**
Quick Reference	**5**
Tutorials	**6**
Reference Part - Overview	**7**
List of Abbreviations	**8**
Contact Information	**9**

Safety Guidelines

This document contains notices which you should observe to ensure your own personal safety as well as to avoid property damage. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol

Danger

Indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

Warning

Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

Caution

Used with the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

Notice

Used without the safety alert symbol indicates a potential situation which, if not avoided, may result in an undesirable result or state.

When several danger levels apply, the notices of the highest level (lower number) are always displayed. If a notice refers to personal damages with the safety alert symbol, then another notice may be added warning of property damage.

Qualified Personnel

The device/system may only be set up and operated in conjunction with this documentation. Only qualified personnel should be allowed to install and work on the equipment. Qualified persons are defined as persons who are authorized to commission, to earth, and to tag circuits, equipment and systems in accordance with established safety practices and standards.

Intended Use

Please note the following:

Warning

This device and its components may only be used for the applications described in the catalog or technical description, and only in connection with devices or components from other manufacturers approved or recommended by Siemens. This product can only function correctly and safely if it is transported, stored, set up and installed correctly, and operated and maintained as recommended.

Trademarks

All designations marked with ® are registered trademarks of Siemens AG. Other designations in this documentation might be trademarks which, if used by third parties for their purposes, might infringe upon the rights of the proprietors.

Copyright Siemens AG 2014. All rights reserved.

Reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Disclaimer of Liability

We have checked the contents of this document for agreement with the hardware and software described. Since deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in the manual are reviewed regularly, and any necessary corrections will be included in subsequent editions. Suggestions for improvement are welcomed.

Siemens AG
Industry Sector
P.O. Box 4848
90327 Nuremberg
Germany

Siemens AG 2014
Technical data subject to change

Table of Contents

1 Preface 18
1.1 Purpose of this Document 18
1.2 Validity of this Document 18
1.3 Audience 18
1.4 Notations 19
2 Scope of Delivery 20
2.1 Scope of Delivery 20
2.2 Unpacking and checking 20
3 Product Characteristics 21
3.1 Introduction 21
3.1.1 What is CMS X-Tools? 21
3.1.2 Software Structure 21
3.1.2.1 General 21
3.1.2.1.1 Executables 21
3.1.2.1.2 X-Tools Client 22
3.1.2.1.3 X-Tools Server 22
3.1.2.1.4 X-Tools Tray 22
3.1.2.1.5 X-Tools Service 22
3.1.2.1.6 Interoperability 22
3.1.2.2 MDS Explorer - Master Data System Explorer 22
3.1.2.3 MMS - Main Management System 23
3.1.2.4 DMS - Device Management System 23
3.1.2.5 MTS - Monitoring System 23
3.1.2.6 ANS - Analyzing System 24
3.1.2.7 STS - Storage System 24
3.2 Hard- and Software Requirements 25
3.2.1 Hardware 25
3.2.2 Software 25
3.3 Editions of X-Tools 26
3.4 Order Numbers 28
4 Installation 29
4.1 Software Installation 29
4.1.1 Install X-Tools and the System Documentation 29
4.1.2 Create Shortcuts on the Desktop 33
4.1.3 Migration from earlier Versions of X-Tools 34
4.1.3.1 Overview 34
4.1.3.2 New IEEE1394 Driver in V 03.03 34
4.1.3.3 Update of Analyzing Models 35
4.1.3.4 Update from V 03.01 SP1 or earlier to V 04.02 35
4.1.3.4.1 Parallel Installation to the old Version 35

4.1.3.5 Update from V 03.02 or later to V 04.02 36
4.1.3.5.1 Overwriting of the old Version 36
4.1.3.5.2 Parallel Installation to the old Version 36
4.1.4 Updates 37
4.2 Hardware Installation 37
4.2.1 How to connect CMS to your Application 37
4.2.2 Communication Interfaces 37
4.2.3 How to connect IEEE1394 Devices to the PC 38
4.2.4 How to connect Ethernet IONs to the PC 38
4.2.5 How to connect USB CTNs to the PC 39
4.2.6 How to connect OPC UA Servers to the PC 39
4.2.7 How to connect OPC UA Clients to the PC 39
4.2.8 Video Card Driver 40
4.2.9 Minimal Firmware Versions 40
5 Quick Reference 41
6 Tutorials 42
6.1 General Description 42
6.2 X-Tools Basics 42
6.2.1 General Description 42
6.2.2 Running X-Tools for the first Time 42
6.2.3 X-Tools User Interface 43
6.2.4 Available Data 45
6.2.5 Configuration Files 46
6.3 Client to Server Connectivity 47
6.3.1 General Description 47
6.3.2 Creation of new Main Profiles 48
6.3.3 Opening of existing Main Profiles 49
6.3.4 Starting and Stopping of Main Profiles 50
6.3.5 Securing of the X-Tools Server 51
6.3.6 X-Tools Server Connectivity Dialog 51
6.4 Global Configurations 52
6.4.1 General Description 52
6.4.2 The Global Options Editor 52
6.4.3 The User Accounts Editor 53
6.4.4 The System Log Viewer 55
6.5 Configuration File Locations 56
6.5.1 General Description 56
6.5.2 Adding of a Configuration File Location 56
6.5.3 Removing of a Configuration File Location 57
6.6 Simulated Devices and Interface Profiles 57
6.6.1 General Description 57
6.6.2 Creation of a new Interface Profile for the Simulation Interface 58

6.6.3 Opening of existing Interface Profiles 60
6.6.4 Starting and Stopping of Interface Profiles 61
6.7 Device Configuration and Device Profiles 62
6.7.1 General Description 62
6.7.2 Creation of a new Device Profile for an ION AnalogInput T001 62
6.7.3 Opening of existing Device Profiles 65
6.7.4 Starting and Stopping of Device Profiles 66
6.8 Visualization of Online Data and Monitoring Views 67
6.8.1 General Description 67
6.8.2 Creation of a new Monitoring View 67
6.8.3 Opening of existing Monitoring Views 70
6.8.4 Starting and Stopping of Monitoring Views 71
6.9 Storage of online Data via Storage Profiles 71
6.9.1 General Description 71
6.9.2 Creation of a new Storage Profile for online Data 71
6.9.3 Opening of existing Storage Profiles 73
6.9.4 Starting and Stopping of Storage Profiles 75
6.10 Loading of Data and Loading Profiles 76
6.10.1 General Description 76
6.10.2 Simple Loading of Offline Data 76
6.10.3 Creation of a new Loading Profile 78
6.10.4 Opening of existing Loading Profiles 81
6.10.5 Starting and Stopping of Loading Profiles 82
6.10.6 Advanced Loading of Offline Data 82
6.11 Offline File Locations 85
6.11.1 General Description 85
6.11.2 Adding of an Offline File Location 85
6.11.3 Removing of an Offline File Location 86
6.12 Visualization of offline Data 86
6.12.1 General Description 86
6.12.2 Visualization of offline Data 87
6.13 Analyzing of Data via Analyzing Models 88
6.13.1 General Description 88
6.13.2 Creation of a new Analyzing Model 88
6.13.3 Opening of existing Analyzing Models 91
6.13.4 Starting and Stopping of Analyzing Models 92
6.14 Analyzing of Data via Analyzing Scripts 94
6.14.1 General Description 94
6.14.2 Creation of a new Analyzing Script 94
6.14.3 Opening of existing Analyzing Scripts 97
6.14.4 Starting and Stopping of Analyzing Scripts 98
6.15 User Functions for the Encapsulation of Analysis Tasks 99

6.15.1 General Description 99
6.15.2 Creation of a new User Function 99
6.15.3 Opening of existing User Functions 104
6.15.4 Using of User Functions within Analyzing Models 106
6.15.5 Using of User Functions within Analyzing Scripts 108
6.15.6 Behavior of User Functions 110
6.15.7 Examples of User Functions 111
6.16 Storage of offline Data via Storage Profiles 112
6.16.1 General Description 112
6.16.2 Creation of a new Storage Profile for offline Data 112
6.17 Scalar Data 114
6.17.1 General Description 114
6.17.2 Creation of scalar Data via the MDS Explorer 114
6.17.3 Creation of scalar Data via Analyzing Models and Analyzing Scripts 115
6.17.4 Using of scalar Data within Analyzing Models and Analyzing Scripts 116
6.17.5 Using of Scalar Data for Input Arguments of Class “Parameter” 119
6.17.6 Changing of the Value of a scalar Data via the MTC Mimic Board T001 120
6.18 3D Visualization 121
6.18.1 General Description 121
6.18.2 Visualization of Data in 3D 121
6.19 Safe Mode 123
6.19.1 General Description 123
6.19.2 Important Facts 123
6.19.3 Safe Mode 123
6.19.4 X-Tools Service 124
6.20 Sequence Control 124
6.20.1 General Description 124
6.20.2 Important Facts 124
6.20.3 Provided Analyzing Functions 125
6.20.4 Example 127
6.21 Opening of Monitoring Views via the Command Line 131
6.21.1 General Description 131
6.21.2 Command Line Syntax 131
6.22 Printing of Monitoring Views via the Command Line 132
6.22.1 General Description 132
6.22.2 Command Line Syntax 132
6.23 Storing of Data Snapshots out of the Monitoring System 134
6.23.1 General Description 134
6.23.2 Storing of Data Snapshots 134
6.24 Receiving of Data from standard OPC UA Servers 136
6.24.1 General Description 136
6.24.2 Connecting to an OPC UA Server 136

6.24.3 Configuring of the Data of interest 137
6.24.4 Using of the Data of interest 138
6.25 Providing of Data to standard OPC UA Clients 139
6.25.1 General Description 139
6.25.2 Enabling of the built-in OPC UA Server 139
6.25.3 Using of the built-in OPC UA Server 140
6.26 Creation of Screenshots out of the Monitoring System 140
6.26.1 General Description 140
6.26.2 Printing of Screenshots 140
6.26.3 Saving of Screenshots 141
6.27 Creation of Automatic Reports out of the Monitoring System 141
6.27.1 General Description 141
6.27.2 Important Facts 141
6.27.3 Creation of a Monitoring View for automatic Reports 142
6.27.4 Trigger Examples 144
6.27.5 Example Configuration Files 145
6.28 Creation of Reports of Analyzing Models 146
6.28.1 General Description 146
6.28.2 Important Facts 147
6.28.3 Printing of a Report 147
6.28.4 Saving of a Report 147
6.29 Fast Analysis via the MDS Explorer 148
6.29.1 General Description 148
6.29.2 Creation of a Location for Fast Analysis Scripts 148
6.29.3 Creation of a new Fast Analysis Script 148
6.29.4 Opening of existing Fast Analysis Scripts 149
6.29.5 Using of Fast Analysis Scripts 150
6.30 Calculation of Formulas via the MVE Standard T001 152
6.30.1 General Description 152
6.30.2 Creation of a simple Formula 152
6.30.3 Running of Formulas 153
6.31 Passing of Command Line Parameters via a Web Browser 154
6.31.1 General Description 154
6.31.2 Important Facts 154
6.31.3 Example Web Page 154
6.32 Parameters for Configuration Files 156
6.32.1 General Description 156
6.32.2 Important Facts 156
6.32.3 Provided Parameters Table 156
6.32.4 Device Profiles 157
6.32.5 Monitoring Views 159
6.32.6 Analyzing Models 161

6.32.7 Analyzing Scripts 162
6.32.8 Storage Profiles 164
6.33 Startup Analyzing Scripts 167
6.33.1 General Description 167
6.33.2 Important Facts 167
6.33.3 Startup Analyzing Script 167
6.34 Supported CSV (*.csv) Format for Data 169
6.34.1 General Description 169
6.34.2 Exporting of Data to CSV Files 169
6.34.3 Importing of Data from CSV Files 169
6.34.4 CSV File Format for Data 170
6.35 File Associations 172
6.35.1 General Description 172
6.35.2 Important Facts 172
6.35.3 Opening of Files via the Explorer of Windows 172
7 Attachment A – Best Practices 174
7.1 Inside Operating Systems 174
7.1.1 General 174
7.1.2 Microsoft Windows XP 174
7.1.3 Microsoft Windows Server 2003 174
7.1.4 Microsoft Windows 7 175
7.2 Time Synchronization 176
7.2.1 General 176
7.2.2 Setup a Windows XP Time Server 176
7.2.3 Setup a Windows XP Time Client 176
7.2.4 Setup of a Computer which is Time Client and Time Server simultaneously 177
7.2.5 Microsoft Hotfix 951531 178
7.3 Memory Optimizations 178
7.3.1 General 178
7.3.2 Default Ring Buffer Size 178
7.3.3 Allocation Size 180
7.3.4 Offline Data 180
7.3.5 Dealing with very huge Amounts of Data 180
7.3.6 Memory Management 182
7.4 Measurement Tasks 183
7.4.1 Frequency Analysis 183
7.4.1.1 General 183
7.4.1.2 Analyzing Model 183
7.4.1.3 Example 184
7.4.2 Spectrum Monitoring 185
7.4.2.1 General 185
7.4.2.2 Analyzing Model 185

7.4.2.3 Example 186
7.4.3 Order Analysis 188
7.4.3.1 General 188
7.4.3.2 Analyzing Model 188
7.4.3.3 Example 190
7.4.4 Sequence Control 192
7.4.4.1 General 192
7.4.4.2 Example 192
7.4.5 X/Y Envelopes 195
7.4.5.1 General 195
7.4.5.2 Creation of X/Y Envelopes 195
7.4.5.3 Visualization of X/Y Envelopes 197
7.4.5.4 Monitoring of X/Y Envelopes 199
7.4.5.5 Example 200
7.4.5.6 Summary 200
7.4.6 Basic Spectrum Envelopes 201
7.4.6.1 General 201
7.4.6.2 Creation of Spectrum Envelopes 201
7.4.6.3 Visualization of Spectrum Envelopes 203
7.4.6.4 Monitoring of Spectrum Envelopes 203
7.4.6.5 Example 204
7.4.6.6 Summary 205
7.4.7 Advanced Spectrum Envelopes 206
7.4.7.1 General 206
7.4.7.2 Preparations 206
7.4.7.3 Creation of Spectrum Envelopes 207
7.4.7.4 Monitoring of online Spectrum Envelopes 208
7.4.7.4.1 Example 209
7.4.7.5 Monitoring of online Amplitudes 211
7.4.7.5.1 Example 212
7.4.7.6 Monitoring of offline Data 214
7.4.7.7 Summary 216
7.4.8 Automatic loading of Data 218
7.4.8.1 General 218
7.4.8.2 Basic loading of offline Data 218
7.4.8.3 Loading of offline Data Files from a certain Period of Time 218
7.4.8.4 Loading with Loading Profiles and Loading Modes 219
7.4.8.5 Loading with Loading Profiles and Loading Times 219
7.4.8.6 Loading of the last Minutes of the Data from a Storage Profile 220
7.4.8.7 Loading of the last Files from a Storage Profile (once) 220
7.4.8.8 Loading of the last Files from a Storage Profile (automatically) 221
7.4.8.9 Summary 221

7.4.9 Advanced parallel Storing 222
7.4.9.1 General 222
7.4.9.2 Plant Simulation 223
7.4.9.3 Storage Profile 223
7.4.9.4 Summary 224
7.4.10 Charge based Analysis 224
7.4.10.1 General 224
7.4.10.2 Plant Simulation 224
7.4.10.3 Charge based Analysis 225
7.4.10.4 Summary 226
7.4.11 File Splitting based Analysis 227
7.4.11.1 General 227
7.4.11.2 Input Data 227
7.4.11.3 File Splitting based Analysis 228
7.4.11.4 Summary 229
7.4.12 Passing of Parameters to Analyzing Models 230
7.4.12.1 General 230
7.4.12.2 Analyzing Models 230
7.4.12.3 StartEx() 231
7.4.12.4 Summary 232
7.4.13 Extrapolation 232
7.4.13.1 General 232
7.4.13.2 Extrapolate() 233
7.4.13.3 Detection of Violation 234
7.4.13.4 Summary 235
7.4.14 Data Analysis with Gaps 235
7.4.14.1 General 235
7.4.14.2 Action based Calculation Mode 235
7.4.14.3 Falsified Average 236
7.4.14.4 True Average 237
7.4.14.5 Automatic Analysis 238
7.4.14.6 Summary 238
7.4.15 Output Modes of Output Analyzing Functions 240
7.4.15.1 General 240
7.4.15.2 Analyzing Model 240
7.4.15.3 Analyzing Script 241
7.4.15.4 Summary 242
7.4.16 #include within Analyzing Scripts 242
7.4.16.1 General 242
7.4.16.2 Preparing of Analyzing Scripts which shall be #included later 242
7.4.16.3 #including of Analyzing Scripts 243
7.4.16.4 Summary 245

7.4.17 InConst() and Single Point Parameter Definitions 246
7.4.17.1 General 246
7.4.17.2 User Functions 246
7.4.17.3 Analyzing Models 248
7.4.17.4 Analyzing Scripts 249
7.4.17.5 Summary 250
7.4.18 Strings as Storage ID 251
7.4.18.1 General 251
7.4.18.2 Acquisition of Strings 251
7.4.18.3 Device Profiles 251
7.4.18.4 Analyzing Models 252
7.4.18.5 Storage Profiles 252
7.4.18.6 Offline Data Files 253
7.4.18.7 Summary 253
7.4.19 Performance Monitoring 254
7.4.19.1 General 254
7.4.19.2 Available Performance Data 254
7.4.19.3 Analyzing Scripts 254
7.4.19.4 Analyzing Models 255
7.4.19.5 Storage Profiles 255
7.4.19.6 Monitoring Views 256
7.4.19.7 Summary 256
7.4.20 Editing of Scalar Data.csv 258
7.4.20.1 General 258
7.4.20.2 Import to Microsoft Excel 258
7.4.20.3 Export from Microsoft Excel 260
7.4.21 Massive sequential processing with Scalar Data 262
7.4.21.1 General 262
7.4.21.2 Preparation of an Example File 262
7.4.21.3 Preparation of the Scalar Data 264
7.4.21.4 Viewing, editing, saving and loading of the Parameters 265
7.4.21.5 Run sequential processing via Scalar Data 267
7.4.21.6 Run sequential processing via local Variables 269
7.4.21.7 Additional Notes 270
7.4.21.8 Summary 270
8 Reference Part - Overview 271
8.1 X-Tools Client 271
8.1.1 Controls 271
8.1.1.1 Tables 271
8.1.1.2 Trees 273
8.1.1.3 Menu Bars 274
8.2 X-Tools Server 275

8.2.1 Overview 275
8.2.2 Icon of the X-Tools Server 275
8.2.3 Tool Tip 275
8.2.4 Context Menu 275
8.2.5 X-Tools Server Status Dialog 276
8.3 X-Tools Tray 277
8.3.1 Overview 277
8.4 X-Tools Service 277
8.4.1 Overview 277
8.4.2 Automatic Start of the X-Tools Server during the Startup of Windows 277
8.4.3 Monitoring of the X-Tools Server during its Runtime 277
8.4.4 Stopping of the X-Tools Service 277
9 List of Abbreviations 278
10 Contact Information 279

Table of Figures

*Figure 1: Edition of the **X-Tools Server** within the* **MMS Explorer** 28
*Figure 2: Edition of the **X-Tools Server** within the* **System Log Viewer** 28
*Figure 3: Termination of running Components through the **Setup*** 29
*Figure 4: Welcome Page of the **Setup*** 29
*Figure 5: License Agreement within the **Setup*** 30
*Figure 6: Selection of the Destination Location within the **Setup*** 30
*Figure 7: Selection of the to-be-installed Components within the **Setup*** 31
*Figure 8: Selection of the additional Tasks which shall be performed by the **Setup*** 31
Figure 9: Overview about the Installation Tasks 32
Figure 10: Overview about the Progress of the Installation 32
Figure 11: Installation completed 33
*Figure 12: Main Title Bar of the **X-Tools Client*** 43
*Figure 13: Main Menu Bar of the **X-Tools Client*** 43
*Figure 14: Main Status Bar of the **X-Tools Client*** 44
Figure 15: Full Screen Button 44
Figure 16: Restore Button 44
Figure 17: Maximize Button 44
Figure 18: Close Button 44
*Figure 19: **MPE Standard T001** with a new Main Profile* 48
*Figure 20: **MPE Standard T001** with an opened Main Profile* 49
*Figure 21: **X-Tools Server Connectivity** dialog* 51
*Figure 22: **IPE Simulation T001** with a new Interface Profile* 58
*Figure 23: **IPE Simulation T001** with an opened Interface Profile* 60
*Figure 24: **DPE ION AnalogInput T001** with a new Device Profile* 63
*Figure 25: **DPE ION AnalogInput T001** with an opened Device Profile* 65
*Figure 26: **MVE Standard T001** with a new Monitoring View* 68
*Figure 27: **MVE Standard T001** with an opened Monitoring View* 70
*Figure 28: **SPE Binary T001** with a new Storage Profile* 72
*Figure 29: **SPE Binary T001** with an opened Storage Profile* 74
*Figure 30: **ODL Standard T001** with some Offline Data Files* 77
*Figure 31: **Loading Progress** Dialog* 78
*Figure 32: **LPE Standard T001** with a new Loading Profile* 79
*Figure 33: **LPE Standard T001** with an opened Loading Profile* 81
*Figure 34: **ODL Standard T001** with some Offline Data Files* 83
*Figure 35: **AME Standard T001** with a new Analyzing Model* 89
*Figure 36: **AME Standard T001** with an opened Analyzing Model* 91
*Figure 37: **ASE Standard T001** with a new Analyzing Script* 95
*Figure 38: **ASE Standard T001** with an opened Analyzing Script* 97
*Figure 39: **UFE Standard T001** with a new User Function* 101
*Figure 40: **UFE Standard T001** with the HelloW() Example* 103

*Figure 41: **UFE Standard T001** with the HelloW2() Example* 104
*Figure 42: **UFE Standard T001** with an opened User Function* 105
*Figure 43: **AME Standard T001** with the HelloW() User Function* 106
*Figure 44: **AME Standard T001** with the HelloW2() User Function* 107
*Figure 45: **ASE Standard T001** with the HelloW() User Function* 108
*Figure 46: **ASE Standard T001** with the HelloW2() User Function* 109
*Figure 47: **SPE Binary T001** with a new Storage Profile* 113
Figure 48: The Analyzing Model "Tutorial\Scalar Data\Triangle" from the Example Location 115
Figure 49: The Analyzing Script "Tutorial\Scalar Data\Triangle" from the Example Location 115
Figure 50: The Analyzing Model "Tutorial\Scalar Data\Sine" from the Example Location 116
Figure 51: The Monitoring View "Tutorial\Scalar Data\Triangle + Sine" from the Example Location 117
Figure 52: The Analyzing Script "Tutorial\Scalar Data\Multiplication" from the Example Location 117
Figure 53: The Monitoring View "Tutorial\Scalar Data\Triangle + Multiplication" from the Example Location ... 118
Figure 54: The Analyzing Script "Tutorial\Scalar Data\Create Parameters" from the Example Location 119
Figure 55: The Analyzing Model "Tutorial\Scalar Data\Use Parameters" from the Example Location 119
Figure 56: The Monitoring View "Tutorial\Scalar Data\Spectrum" from the Example Location 120
*Figure 57: **MVE Standard T001** with a 3D Visualization* 122
Figure 58: Example Use Case for the Sequence Control 124
Figure 59: Example for an Automatic Report 146
*Figure 60: **ASE Standard T001** with an opened Fast Analysis Script* 150
Figure 89: Example of a Provided Parameters Table 157
*Figure 89: Example of the Provided Parameters Table of the **DPE IFN ANALOGINPUT*** 157
Figure 90: Example of an Analyzing Script which starts a Device Profile with Parameters 158
Figure 90: Example of an Analyzing Script which creates a new Device Profile out of a Template 158
*Figure 89: Example of the Provided Parameters Table of the **MVE Standard T001*** 159
Figure 90: Example of an Analyzing Script which starts a Monitoring View with Parameters 159
Figure 90: Example of an Analyzing Script which creates a new Monitoring View out of a Template 160
*Figure 89: Example of the Provided Parameters Table of the **AME Standard T001*** 161
Figure 90: Example of an Analyzing Script which starts an Analyzing Model with Parameters 161
Figure 90: Example of an Analyzing Script which creates a new Analyzing Model out of a Template 162
Figure 90: Example of an Analyzing Script which uses a provided Parameter 163
*Figure 89: Example of the Provided Parameters Table of the **ASE Standard T001*** 163
Figure 90: Example of an Analyzing Script which starts an Analyzing Script with Parameters 163
Figure 90: Example of an Analyzing Script which creates a new Analyzing Script out of a Template 164
*Figure 89: Example of the Provided Parameters Table of the **SPE Binary T001*** 165
Figure 90: Example of an Analyzing Script which starts a Storage Profile with Parameters 165
Figure 90: Example of an Analyzing Script which creates a new Storage Profile out of a Template 166
Figure 90: Example of a Startup Analyzing Script 167
Figure 61: Example of an Analyzing Model for a Frequency Analysis 183
Figure 62: Example of the Input Signal for the Frequency Analysis 184
Figure 63: Example of a Frequency Spectrum which has been calculated by the Frequency Analysis 184
Figure 64: Example of an Analyzing Model which monitors a Frequency Spectrum 185

Figure 65: Example of the Input Spectrum for the Spectrum Monitoring 186
Figure 66: Example of the Input Spectrum with Alarm Spectra 187
Figure 67: Example of an Alarm Log Entry 187
Figure 68: Example of the derived Input Spectrum with Alarm Spectra 188
Figure 69: Example of an Analyzing Model for the Order Analysis 189
Figure 70: Example of the Input Signals for the Order Analysis 190
Figure 71: Example of a Frequency Spectrum and of an Order Spectrum 191
Figure 72: Example of the Input Data for an X/Y Envelope 195
Figure 73: Example of the Input Data for an X/Y Envelope together with the calculated Envelopes 198
Figure 74: Example of an Analyzing Model for the monitoring of the current x and y online Data 199
Figure 75: Example of an Analyzing Script for the monitoring of the current x and y online Data 199
Figure 76: Example of the Output of MonitorXYEnvelope() 200
Figure 77: Example of the Input Data for a basic Spectrum Envelope 201
Figure 78: Example of an Analyzing Model for the Creation of Spectrum Envelopes 202
Figure 79: Example of an Analyzing Script for the Creation of Spectrum Envelopes 202
Figure 80: Example of the Input Data for a Spectrum Envelope together with the calculated Envelopes 203
Figure 81: Example of an Analyzing Model for the monitoring of the current online Spectrum 203
Figure 82: Example of an Analyzing Script for the monitoring of the current online Spectrum 204
Figure 83: Example of the Output of MonitorSpectrumEnvelope() 204
Figure 77: Example of the Input Data for an advanced Spectrum Envelope 207
Figure 81: Example of an Analyzing Model for the monitoring of the current online Spectrum 208
Figure 82: Example of an Analyzing Script for the monitoring of the current online Spectrum 208
Figure 83: The Monitoring View "20 View online Violations" with the default Wave 209
Figure 83: The Monitoring View "Online\20 View online Violations" with a modified Wave 210
Figure 81: Example of an Analyzing Model for the monitoring of Amplitudes 211
Figure 82: Example of an Analyzing Script for the monitoring of Amplitudes 211
Figure 83: The Monitoring View "35 View online Amplitude Trends" with the default Wave 212
Figure 83: The Monitoring View "35 View online Amplitude Trends" with a modified Wave 213
Figure 83: The Monitoring View "20 View offline Violations" 214
Figure 83: The Monitoring View "35 View offline Amplitude Trends" 215
Figure 84: Example of the Plant Simulation for the advanced parallel storing 223
Figure 85: Example of the Process Simulation for the Charge based Analysis 225
Figure 86: Example of an Analyzing Script for the Charge-based processing 225
Figure 87: Example of the simulated Data for the File Splitting based Analysis 227
Figure 88: Example of an Analyzing Script for the File Splitting based processing 228
Figure 89: Example of a Provided Parameters Table 231
Figure 90: Example of an Analyzing Script which does not pass any Parameters 231
Figure 91: Example of an Analyzing Script which passes a few Parameters 231
Figure 92: Example of an Analyzing Script which passes all Parameters 232
*Figure 93: Example of an Analyzing Script which uses **Extrapolate()*** 233
*Figure 94: Example Output of **Extrapolate()*** 233
Figure 95: Example of an Analyzing Script which searches for a Violation of the given Threshhold 234

Figure 96: Extrapolated Data together with the detected Violation 234
Figure 97: Analyzing Model which uses the Action based Calculation Mode for calculating of an Average 235
Figure 98: Example of an Average which was calculated via the Action based Calculation Mode 236
Figure 99: Example of a falsified Average 236
Figure 100: Analyzing Model which uses GetStatus() in order to be able to determine the true Average 237
Figure 101: Example of an Average which was calculated with considering of Gaps 237
Figure 102: Analyzing Model which uses different Output Modes 240
Figure 103: Analyzing Script which uses different Output Modes 241
Figure 104: The Analyzing Script "05 Defines" from the Example Location 242
Figure 105: The Analyzing Script "10 Functions" from the Example Location 243
Figure 106: The Analyzing Script "10 Types" from the Example Location 243
Figure 107: Example of a sequential Analyzing Script which uses #include Statements 243
Figure 108: Output of a sequential Analyzing Script which uses #include Statements 244
Figure 109: Example of a Time Slice based Analyzing Script which uses #include Statements 244
Figure 110: Output of a Time Slice based Analyzing Script which uses #include Statements 244
Figure 111: The User Function "Example for InConst()" from the Example Location 246
Figure 112: The Analyzing Model "Example with User Function" from the Example Location 247
Figure 113: Example output of the Monitoring View "Example with User Function" 247
Figure 114: The Analyzing Model "Example for InConst()" from the Example Location 248
Figure 115: The Analyzing Script "Example with Analyzing Model" from the Example Location 248
Figure 116: Example output of the Monitoring View "Example for InConst()" from the Example Location 248
Figure 117: The Analyzing Script "Example for const" from the Example Location 249
Figure 118: Example output of the Monitoring View "Example for const" from the Example Location 249
Figure 119: The Device Profile "ION SIMATIC S7 PN" from the Example Location 251
Figure 120: The Analyzing Model "Storage ID" from the Example Location 252
Figure 121: The Storage Profile "Storage ID" from the Example Location 252
Figure 122: Example for the Offline Data Files which are created by the Storage Profile "Storage ID" 253
Figure 123: The Analyzing Script "Start Acquisition\System" from the Example Location 254
Figure 124: The Analyzing Script "Stop Acquisition\System" from the Example Location 255
Figure 125: The Analyzing Model "X-Tools Server\MonFrBlck()" from the Example Location 255
Figure 126: Example output of the Monitoring View "X-Tools Server" 256
Figure 127: Import via Microsoft Excel - Step 1 258
Figure 128: Import via Microsoft Excel - Step 2 259
Figure 129: Import via Microsoft Excel - Step 3 259
Figure 130: Import via Microsoft Excel - Step 4 260
Figure 131: Import via Microsoft Excel - Step 5 260
Figure 132: Export via Microsoft Excel - Step 1 260
Figure 133: Export via Microsoft Excel - Step 2 261
Figure 134: Export via Microsoft Excel - Step 3 261
Figure 135: The Analyzing Script "10 Prepare File" from the Example Location 262
Figure 136: The Analyzing Script "00 Helpers" from the Example Location 263
Figure 137: Contents of "C:\Temp\Default.ini" after running of the Analyzing Script "10 Prepare File" 264

Figure 138: The Analyzing Script “10 Prepare Scalar Data” from the Example Location 264
Figure 139: The Monitoring View “Configure Measurement Points” from the Example Location 265
Figure 140: The Analyzing Script “20 Configure Measurement Points” from the Example Location 266
Figure 141: The Analyzing Script “50 Run sequential processing via Scalar Data” from the Example Location 267
Figure 142: The Analyzing Script “50 Run sequential processing via Scalar Data” from the Example Location 268
Figure 143: The Analyzing Script “50 Run sequential processing via local Variables” from the Example Location 269
*Figure 144: Example of a Tool Tip of the **X-Tools Server*** 275
*Figure 145: **X-Tools Server Status** Dialog* 276

1 Preface

1.1 Purpose of this Document

This document provides an introduction about the functionalities and usage of the software

- CMS X-Tools

of the CMS product line.

In addition to the introduction which is found within this document and which provides an overview about the available functionality, the following documentation is available:

- CMS X-Tools - User Manual - 02 - Master Data System
  - provides detailed information about the functionality which is provided by the **Master Data System**
- CMS X-Tools - User Manual - 03 - Main Management System
  - provides detailed information about the functionality which is provided by the **Main Management System**
- CMS X-Tools - User Manual - 04 - Device Management System
  - provides detailed information about the functionality which is provided by the **Device Management System**
- CMS X-Tools - User Manual - 05 - Monitoring System
  - provides detailed information about the functionality which is provided by the **Monitoring System**
- CMS X-Tools - User Manual - 06 - Analyzing System
  - provides detailed information about the functionality which is provided by the **Analyzing System**
- CMS X-Tools - User Manual - 07 - Storage System
  - provides detailed information about the functionality which is provided by the **Storage System**
- CMS X-Tools - Release Notes
  - provides additional information about the released version of CMS ***X-Tools***
- CMS X-Tools - Change Log
  - provides an overview about the changes which have been introduced with the current version of CMS ***X-Tools***

1.2 Validity of this Document

This document is valid for the following software:

- CMS X-Tools Pro V 04.02
- CMS X-Tools ODV V 04.02

During the following pages, these software packages will be referred to by the term ***X-Tools***. Not all editions of ***X-Tools*** provide all functionality, thus it may be that some of the descriptions within the user manual do not apply to all editions.

1.3 Audience

This document is intended for personnel involved in the commissioning and using of the software:

- ***X-Tools***

1.4 Notations

The following notations are used within this document:

- ***bold, italic*** text is being used for the main executables of ***X-Tools***
  - examples: ***X-Tools Client***, ***X-Tools Server***
- **bold** text is being used for the software modules of ***X-Tools***
  - examples: **Main Management System**, **Device Profile Editor**, **IPE Socket T001**
- green text is being used for controls like tables and trees
  - examples: Main Profile Settings table, Device Profile Data table
- orange text is being used for simple controls like a menu button, a single row/column/cell of a table or a branch of a tree
  - examples: Open… menu button, IP Address column, Target Device Name cell, Interfaces Branch
- dark yellow text is being used for the entries of context menus
  - examples: Advanced Append…, Edit
- Camel Notation is being used for major terms of ***X-Tools***
  - examples: Main Profile, Interface Profile, User Accounts File, Analyzing Function
- < and > brackets are being used for keyboard keys
  - examples: <Ctrl>, <Alt>, <Shift>, <Del>
- [and] brackets are being used for mouse operations
  - examples: [left mouse button down], [mouse move]

2 Scope of Delivery

2.1 Scope of Delivery

What is shipped?

- CD
  - Setup
  - CMS X-Tools - Software & Documentation
  - Software I/O-Nodes - Software & Documentation
    - ION SIMATIC S7-300 T001
    - ION SIMATIC S7-400 T001
    - ION SIMATIC S7-1200
    - ION SIMATIC S7-1500
    - ION SIMATIC S7 PN
    - ION SIMATIC TDC T001
    - ION SIMOTION T001
- Certificate of License
- USB Stick which contains the license information (“Professional” edition only)

2.2 Unpacking and checking

After unpacking, please check

- the packet for completeness and
- all parts for transport damage.

Notice

Do not use any parts that show evidence of damage!

3 Product Characteristics

3.1 Introduction

3.1.1 What is CMS X-Tools?

CMS ***X-Tools*** is a cost-effective, modular, PC-based measurement system, which has been specially optimized to perform reactionless measurement of analog, binary and numerical signals in industrial plants. It consists out of a PC-based software (CMS ***X-Tools***) and various types of measurement devices (***IONs***, ***IFNs*** and ***CTNs***)

With a sampling rate of up to 192 kHz, even highly dynamic processes and occurrences can be recorded with high resolution. The system is used as a portable and/or installed unit in the fields of research & development, testing, error tracking and production.

Input and output signals are handled by individual devices. To accommodate a higher number of signals, basically up to 62 devices can be networked over the IEEE1394 bus system and can be connected to any PC, laptop or server with an IEEE1394 interface.

Despite the considerable features of the system, an experienced technician, with no prior knowledge about CMS ***X-Tools***, can carry out measurements within a very short period of time. Simple measurement tasks, such as paperless recording of plant signals or fault recording, are quickly and effectively carried out.

For higher sophisticated measurements, such as remote service & maintenance or condition monitoring, a high level of graphical configuration and parameterization is possible within CMS ***X-Tools***. In addition, the powerful trigger machine and the analyzing engine enable the user to perform more extensive measurement tasks.

The following topics provide a short overview about the functionalities of the main parts of the CMS ***X-Tools*** software.

3.1.2 Software Structure

3.1.2.1 General

3.1.2.1.1 Executables

X-Tools is a client - server application which consists out of the following executables:

- “Professional” edition:
  - X-Tools Client Pro.exe
  - X-Tools Server Pro.exe
  - X-Tools Tray Pro.exe
  - X-Tools Service Pro.exe
- “Offline Data Viewer” edition:
  - X-Tools Client ODV.exe
  - X-Tools Server ODV.exe
  - X-Tools Tray ODV.exe

The client and the server communicate with each other via standard mechanisms of TCP/IP, thus they can be running either on the same computer or on any two computers which are connected via TCP/IP.

In the default configuration it is not necessary to deal with the details of the client - server architecture at all. Instead it is enough to start the ***X-Tools Client***, in which case the ***X-Tools Server*** is started automatically and also the connection from the ***X-Tools Client*** to the ***X-Tools Server*** is being established automatically. Thus, the ***X-Tools Client*** and the ***X-Tools Server*** behave very much like a single application whenever the additional advantages of the client - server architecture are not needed.

3.1.2.1.2 X-Tools Client

The ***X-Tools Client*** provides the user interface which is being used in order to configure any of the currently connected ***X-Tools Servers***. The ***X-Tools Client*** itself is being used exclusively for the configuration of the measurement system and for the visualization of the present online and offline data. All of the actual measurement and analysis tasks are being processed by the ***X-Tools Server*** according to the made configuration.

Each ***X-Tools Client*** can connect to multiple ***X-Tools Servers*** at once in order to configure it and/or to visualize its present online and offline data.

See point 8.1 for more details about the ***X-Tools Client***.

3.1.2.1.3 X-Tools Server

The ***X-Tools Server*** does not provide any user interface and is fully configured via one or more of the currently connected ***X-Tools Clients***. In case there is not any ***X-Tools Client*** attached to the ***X-Tools Server*** at the moment, it still continues to perform all of its measurement, analysis and storage tasks.

Depending to the edition of the ***X-Tools Server*** (Demo, Standard or Professional), up to 16 clients can be connected simultaneously and access the Configuration Files of the ***X-Tools Server*** as well as its available online and offline data. See point 3.3 for a detailed overview about the different capabilities of the available editions of the ***X-Tools Server***.

See point 8.2 for more details about the ***X-Tools Server***.

3.1.2.1.4 X-Tools Tray

The ***X-Tools Tray*** provides the tray icon which shows basic information about the ***X-Tools Server***. It allows to acces this basic information from multiple Windows sessions simultaneously.

See point 8.3 for more details about the ***X-Tools Tray***.

3.1.2.1.5 X-Tools Service

The ***X-Tools Service*** is not being accessed directly. Its task is to monitor the ***X-Tools Server*** during its operation in order to provide the mechanisms of the Safe Mode (see point 6.19) and to start the ***X-Tools Server*** in case it is configured to be started before a user logs in to Windows.

See point 8.4 for more details about the ***X-Tools Service***.

3.1.2.1.6 Interoperability

Each executable of ***X-Tools*** communicates with another executable of ***X-Tools*** only in case their major and minor versions are identical.

3.1.2.2 MDS Explorer - Master Data System Explorer

- Provides information about all of the currently available data:
  - Online Data
  - Offline Data
  - Scalar Data
- The **MDS Explorer** (short name **MDX**) is the source of all data based operations. In case any data shall be used within another module of ***X-Tools*** (**DMS**, **MTS**, **ANS**, **STS** or any of their sub modules), the desired data can be dragged into this module from the ***MDX***.

3.1.2.3 MMS - Main Management System

- Management of the connections to ***X-Tools Servers*** via Main Profiles
- Configuration of the Global Options of the ***X-Tools Client*** via the **Global Options Editor**
- Viewing of the System Log of the ***X-Tools Client*** via the **System Log Viewer**
- Configuration of the Global Options of any connected ***X-Tools Server*** via the **Global Options Editor**
- Viewing of the System Log of any connected ***X-Tools Server*** via the **System Log Viewer**
- Configuration of the User Accounts of any connected ***X-Tools Server*** via the **User Accounts Editor**
- Viewing of the currently available items via the **Main Management System Explorer**:
  - available Main Profile Editors
  - available Main Profile Files
  - connected ***X-Tools Servers***

3.1.2.4 DMS - Device Management System

- Management of the connections to devices (***IONs***, ***IFNs*** and ***CTNs***) via Interface Profiles
- Configuration of the measurement tasks of each device via Device Profiles
- Viewing of the currently available items via the **Device Management System Explorer**:
  - available **Interface Profile Editors**
  - available **Device Profile Editors**
  - connected ***X-Tools Servers***
    - for each connected ***X-Tools Server***, the following information is available:
    - available Interface Profile Files
    - available Device Profile Files
    - available Device Setup Files
    - available interfaces
    - connected devices
    - available device specific plug-ins (Device Process Modules)

3.1.2.5 MTS - Monitoring System

- Visualization of online and offline data via **Monitoring Views** and **Monitoring Charts**
- Viewing of the currently available items via the **Monitoring System Explorer**:
  - available **Monitoring Charts**
  - available **Monitoring Views**
  - connected ***X-Tools Servers***
    - for each connected ***X-Tools Server***, the following information is available:
    - available Monitoring View Files
    - available visualization specific plug-ins (Monitoring Process Modules)
    - available Analyzing Functions
    - available User Functions

3.1.2.6 ANS - Analyzing System

- Graphical configuration of analysis tasks for online and offline data calculations via Analyzing Models
- Textual configuration of analysis tasks for online and offline data calculations via Analyzing Scripts
- Encapsulation of analysis tasks for online and offline data calculations via User Functions
- Viewing of the currently available items via the **Analyzing System Explorer**:
  - available **Analyzing Model Editors**
  - available **Analyzing Script Editors**
  - available **User Function Editors**
  - connected ***X-Tools Servers***
    - for each connected ***X-Tools Server***, the following information is available:
    - available Analyzing Model Files
    - available Analyzing Script Files
    - available User Function Files
    - available Analyzing Functions
    - available User Functions

3.1.2.7 STS - Storage System

- Configuration of storage tasks via Storage Profiles
- Configuration of loading tasks via Loading Profiles
- Loading of offline data via **Offline Data Loaders**
- Viewing of the currently available items via the **Storage System Explorer**:
  - available **Storage Profile Editors**
  - available **Loading Profile Editors**
  - available **Offline Data Loaders**
  - connected ***X-Tools Servers***
    - for each connected ***X-Tools Server***, the following information is available:
    - available Storage Profile Files
    - available Loading Profile Files
    - available storage format specific plug-ins (Storage Process Modules)

3.2 Hard- and Software Requirements

3.2.1 Hardware

- PC with Windows XP Professional 32-Bit Service Pack 3 *or*
- PC with Windows XP Embedded 32-Bit Service Pack 2 or higher *or*
- PC with Windows 7 Professional 32-Bit Service Pack 1 *or*
- PC with Windows 7 Professional 64-Bit Service Pack 1 *or*
- PC with Windows Server 2008 R2 Enterprise 64-Bit Service Pack 1
- PC with Windows Server 2012 R2 64-Bit
  - Microsoft Hotfix 958347 installed (Windows XP only - see release notes for details)
  - Microsoft Hotfix 951531 installed (Windows XP and Windows Server 2003 - only in case the Windows time service is being used - see point 0 for details)
  - CPU with ≥ 1,2 GHz or multi core CPU
  - working memory ≥ 512 MByte (2 GByte or more recommended)
  - OpenGL and/or DirectX enabled graphic controller (from 1024x768 Pixels, 1280x1024 Pixels or more recommended)
  - IEEE1394, Ethernet and USB interfaces
- power supply units and connecting cables (depending on the used CMS components)

3.2.2 Software

- CMS X-Tools (arbitrary edition)

Note

The usage of resource-intensive applications (e.g. remote-, backup- and/or anti-virus-software, compiler, …) simultaneously to ***X-Tools*** can cause that there are not enough system resources (working memory, CPU time, graphics power, disk throughput, network throughput, …) available for the execution of ***X-Tools***. As tasks like the acquiring of measurement values can be noticeable affected by low available system resources, the available system resources must not be claimed entirely by other applications during the execution of ***X-Tools***.

The exact amount of resources (working memory, CPU time, graphics power, disk throughput, network throughput, …), which are being needed by ***X-Tools*** is dependent to the amount of attached devices and to the amount and size of the configured tasks. The more data is to be acquired by ***X-Tools*** simultaneously and the more other tasks (analysis, storing, visualization, …) are to be performed in parallel, the more resources must be provided by the computer which is running ***X-Tools***. In case of system overload, proper system behavior can not be ensured.

Note

In case the ***X-Tools Client*** and the ***X-Tools Server*** are running at different computers, it is recommended to synchronize the system clocks of the used computers. As ***X-Tools*** does not use an internal mechanism for the time synchronization between ***X-Tools Client*** and ***X-Tools Server***, the visualization of data at the ***X-Tools Client*** may be shifted to the past or to the future in case its system time is not synchronized to the ***X-Tools Server*** that provides the visualized data.

The used method for the time synchronization must ensure that the system time of the synchronized computers does not produce noticeable jumps into the past or into the future. See point 7.2 for general information about time synchronization mechanisms which are built-in to Windows.

Note

In case the ***X-Tools Client*** and the ***X-Tools Server*** are running at different computers, the used connection must provide a throughput of at least 64 kBit/s and a latency of less than two seconds.

Weaker connections can cause errors during the connecting process and during operations like opening of Configuration Files, closing of Configuration Files and visualization of data.

For fluent handling of big files and lots of visualizations, a throughput of above 512 kBit/s and a latency of less than one second are recommended.

Note

Various system configurations have been tested on SIMATIC RACK PCs and SIMATIC MICROBOX PCs. Other standard PC configurations also have been tested successfully, but proper system behavior only can be ensured on SIMATIC platforms.

At the moment of the release of V 04.02 of ***X-Tools***, only the C-series of SIMATIC PCs was released for Microsoft Windows 7. Therefore, proper system behavior of all other series of SIMATIC PCs together with Microsoft Windows 7 can not be ensured.

Note

The Configuration Files and historical data of ***X-Tools*** versions before V 03.00 can not be used by ***X-Tools*** V 04.02 directly. Detailed information about how to migrate from earlier versions is obtained from your local Siemens office and from the homepage http://www.siemens.com/siplus-cms.

Note

In case of a power supply crash some data can be lost. We strongly advise to power the ***X-Tools*** PC with a redundant powersupply or a UPS.

Note

In case of a disk crash some data can be lost. We strongly advise to set up the ***X-Tools*** PC with a RAID disk system or another redundant storage.

3.3 Editions of X-Tools

There are two editions of the ***X-Tools Server*** available: “Professional” and “Offline Data Viewer”. The main functionalities of these editions are:

Edition	Remarks
CMS X-Tools Professional	supports all of the functionality which is described within this user manual
CMS X-Tools Offline Data viewer	the **Analyzing System** workspace is not available the **Device Management System** workspace is not available user accounts are not available Fast Analysis Scripts can be started via the **MDS Explorer** the Formula Editor from the **MVE Standard T001** can be used in roder to configure and run calculations the interfaces to devices and to the OPC UA server are not available it is not possible to start Interface Profiles, Device Profiles or Analyzing Models it is not possible to create any kind of online data the ***X-Tools Service*** is not supported

Each edition is available as 32 bit and as 64 bit application.

One (but not more) instance of the ***X-Tools Server*** of the “Offline Data Viewer” edition can be started in parallel to one (but not more) instance of the ***X-Tools Server*** of “Professional” edition.

The ***X-Tools Client*** from the “Professional” edition can connect to both editions of the ***X-Tools Server***. Contrary, the ***X-Tools Client*** from the “Offline Data Viewer” edition can connect only to an ***X-Tools Server*** which is also an “Offline Data Viewer” edition.

During its startup, the ***X-Tools Server*** from the “Professional” edition checks the present license keys and starts up accordingly. The ***X-Tools Server*** from the “Offline Data Viewer” edition does not perform any license checks.

The following table shows the correlation between license keys and the according functionality of the ***X-Tools Server*** from the “Professional” edition:

License Key	Provided Functionality
CMS X-Tools Professional	This license is needed in order to start up with the full functionality of the “Professional” edition. In case the license is not present, the “Professional” edition terminates automatically about 1 hour after the startup of Windows.
CMS X-Tools Analyzing Bundle 1	This license is needed in order to use the Analyzing Functions from the “Analyzing Bundle 1”.

The current edition of the ***X-Tools Server*** can be seen within the **MMS Explorer** below “MMS Explorer -> Servers -> YourServer -> Info” and within the **System Log Viewer**.

*Figure 1: Edition of the **X-Tools Server** within the* **MMS Explorer**

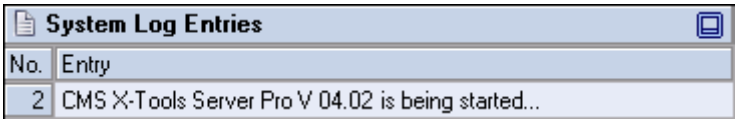


*Figure 2: Edition of the **X-Tools Server** within the* **System Log Viewer**

The whole license key management is handled by the software “Automation License Manager”. It is a standard Siemens product and needs to be installed on the local computer. In case it is not present yet, the Automation License Manager is installed automatically through the setup of the “Professional” edition of ***X-Tools***.

The Automation License Manager contains the functionalities for transferring, saving and checking of license keys. For further information about the Automation License Manager, the included HTML and context-sensitive help can be used.

3.4 Order Numbers

Order information is obtained from your local Siemens office and from the homepage http://www.siemens.com/siplus-cms.

4 Installation

4.1 Software Installation

4.1.1 Install X-Tools and the System Documentation

Note

In order to run ***Setup*** and to install all of the necessary packages, administrator rights are required. Therefore it must be ensured that administrator rights are granted to the Windows user who starts the ***Setup***.

Start the ***Setup*** from the installation media: “Setup.exe”

In case there are components of a previous installation of ***X-Tools*** running already, ***Setup*** asks whether it shall terminate these components:

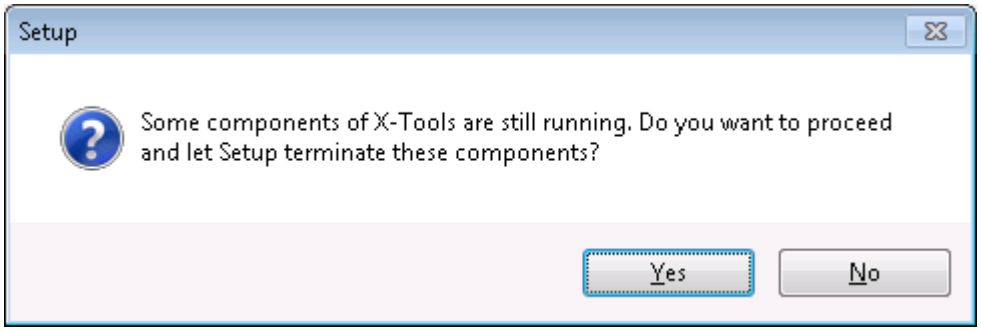

Setup
Some components of X-Tools are still running. Do you want to proceed and let Setup terminate these components?
Yes
No

*Figure 3: Termination of running Components through the **Setup***

The ***Setup*** can not continue while there is any component of ***X-Tools*** running.

The welcome page informs about the variant of ***X-Tools*** which is going to be installed:

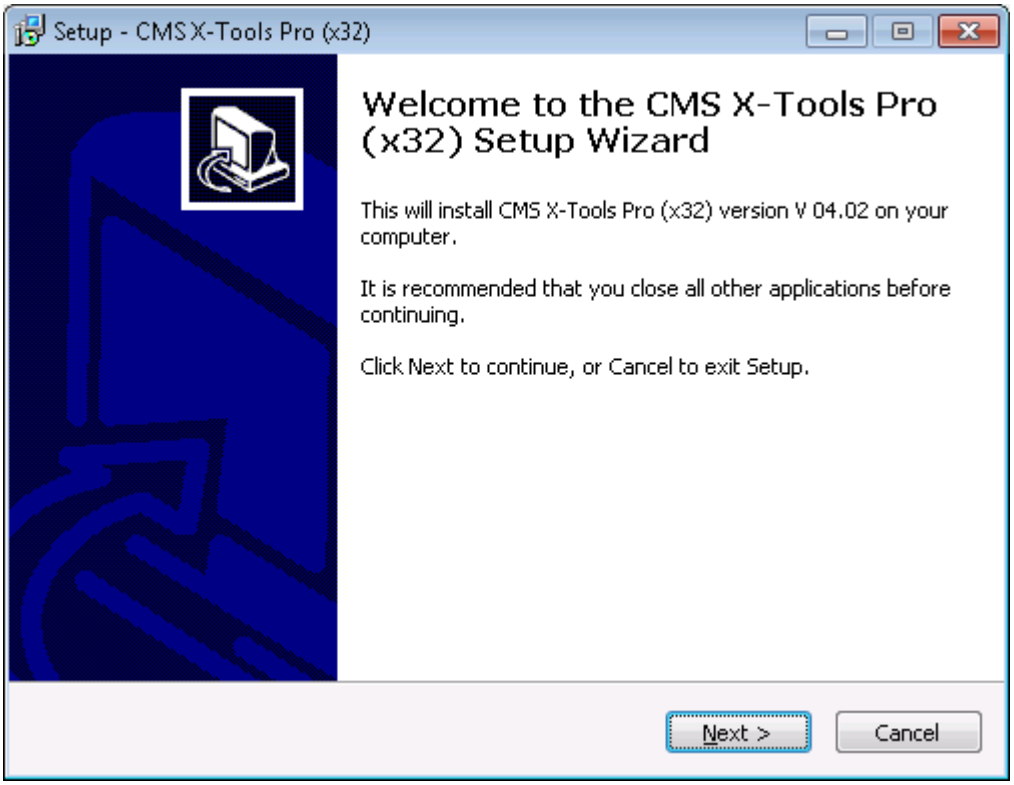

Setup - CMS X-Tools Pro (x32)
Welcome to the CMS X-Tools Pro (x32) Setup Wizard
This will install CMS X-Tools Pro (x32) version V 04.02 on your computer.
It is recommended that you close all other applications before continuing.
Click Next to continue, or Cancel to exit Setup.
Next >
Cancel

*Figure 4: Welcome Page of the **Setup***

Read the license agreement at the next page of the ***Setup*** and choose to accept it:

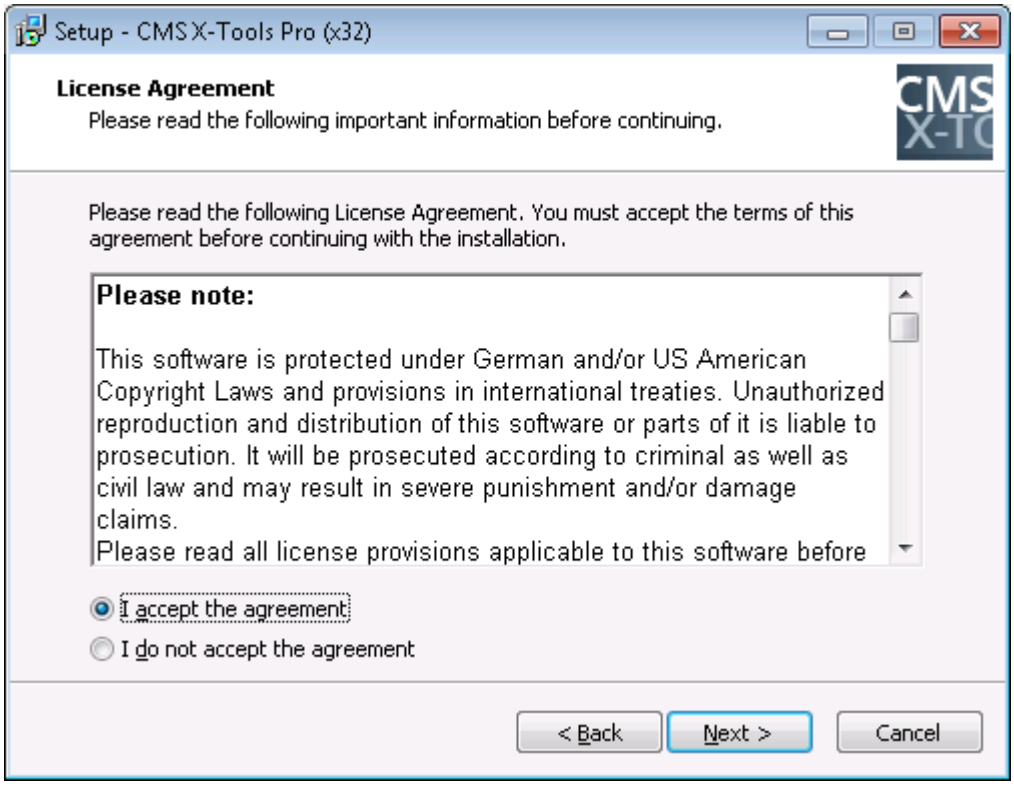

Setup - CMS X-Tools Pro (x32)
License Agreement
Please read the following important information before continuing.
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.
Please note:
This software is protected under German and/or US American Copyright Laws and provisions in international treaties. Unauthorized reproduction and distribution of this software or parts of it is liable to prosecution. It will be prosecuted according to criminal as well as civil law and may result in severe punishment and/or damage claims.
Please read all license provisions applicable to this software before
I accept the agreement
I do not accept the agreement
< Back
Next >
Cancel

*Figure 5: License Agreement within the **Setup***

Choose the desired installation folder within the next page of the ***Setup***:

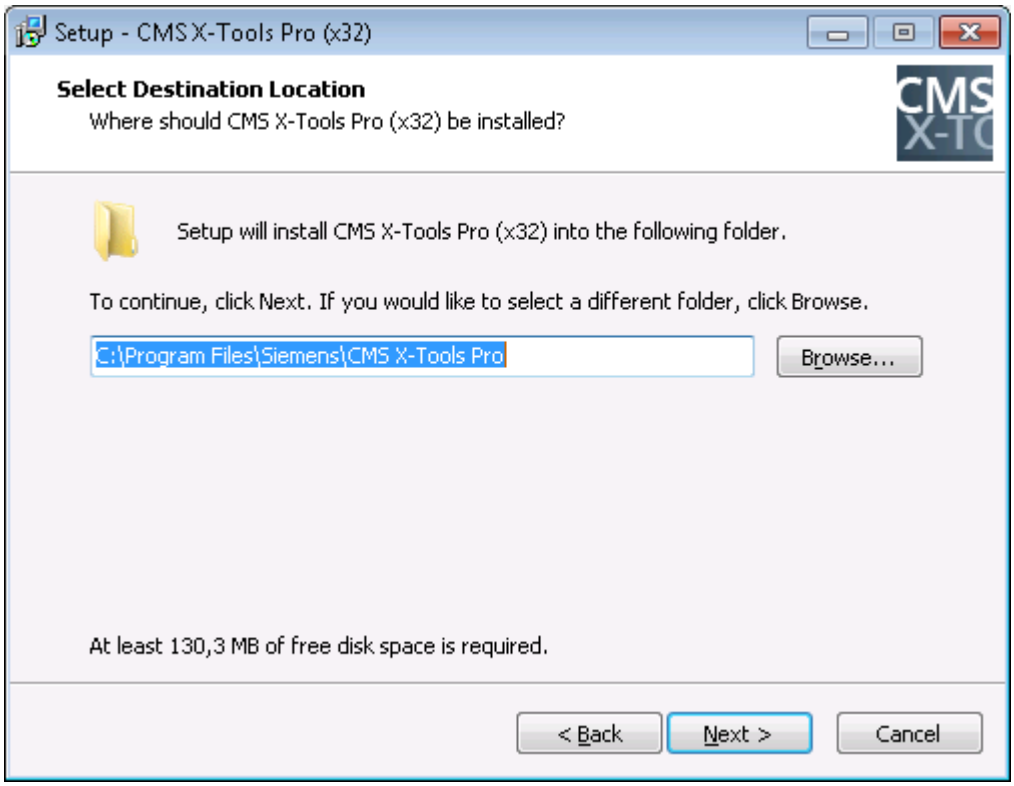

Setup - CMS X-Tools Pro (x32)
Select Destination Location
Where should CMS X-Tools Pro (x32) be installed?
Setup will install CMS X-Tools Pro (x32) into the following folder.
To continue, click Next. If you would like to select a different folder, click Browse.
C:\Program Files\Siemens\CMS X-Tools Pro
Browse...
At least 130,3 MB of free disk space is required.
< Back
Next >
Cancel

*Figure 6: Selection of the Destination Location within the **Setup***

Select the components of ***X-Tools*** which shall be installed:

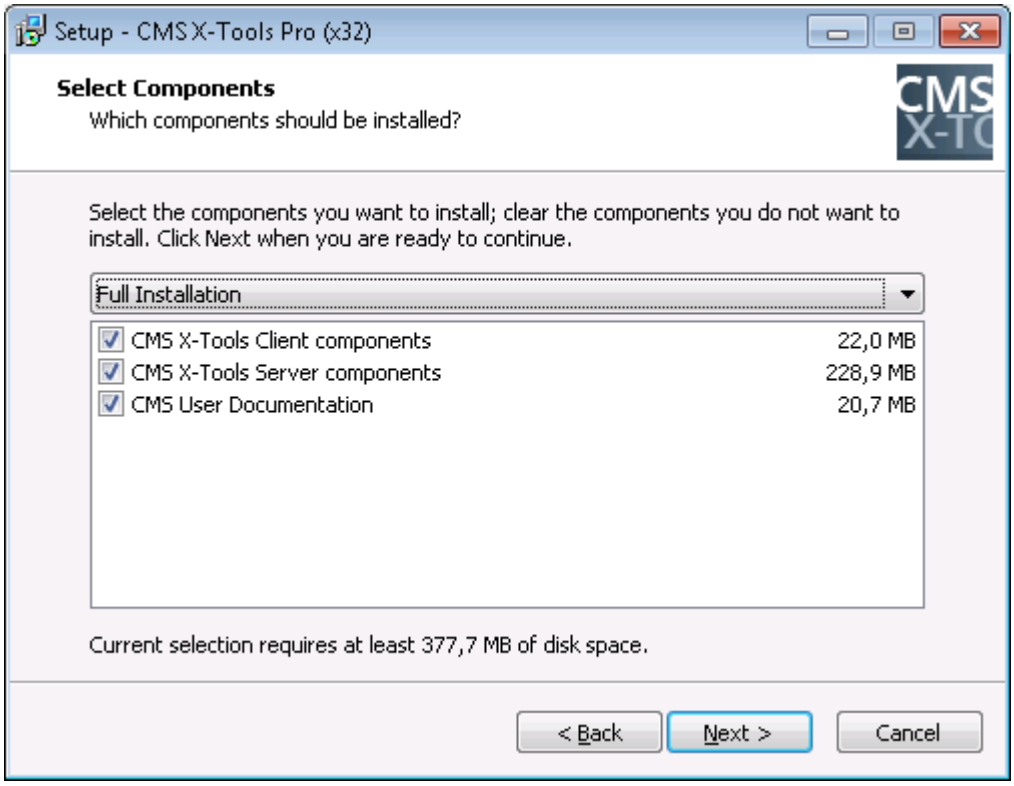

Setup - CMS X-Tools Pro (x32)
Select Components
Which components should be installed?
Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.
Full Installation
CMS X-Tools Client components 22,0 MB
CMS X-Tools Server components 228,9 MB
CMS User Documentation 20,7 MB
Current selection requires at least 377,7 MB of disk space.
< Back
Next >
Cancel

*Figure 7: Selection of the to-be-installed Components within the **Setup***

Select the additional tasks which shall be performed by the ***Setup***:

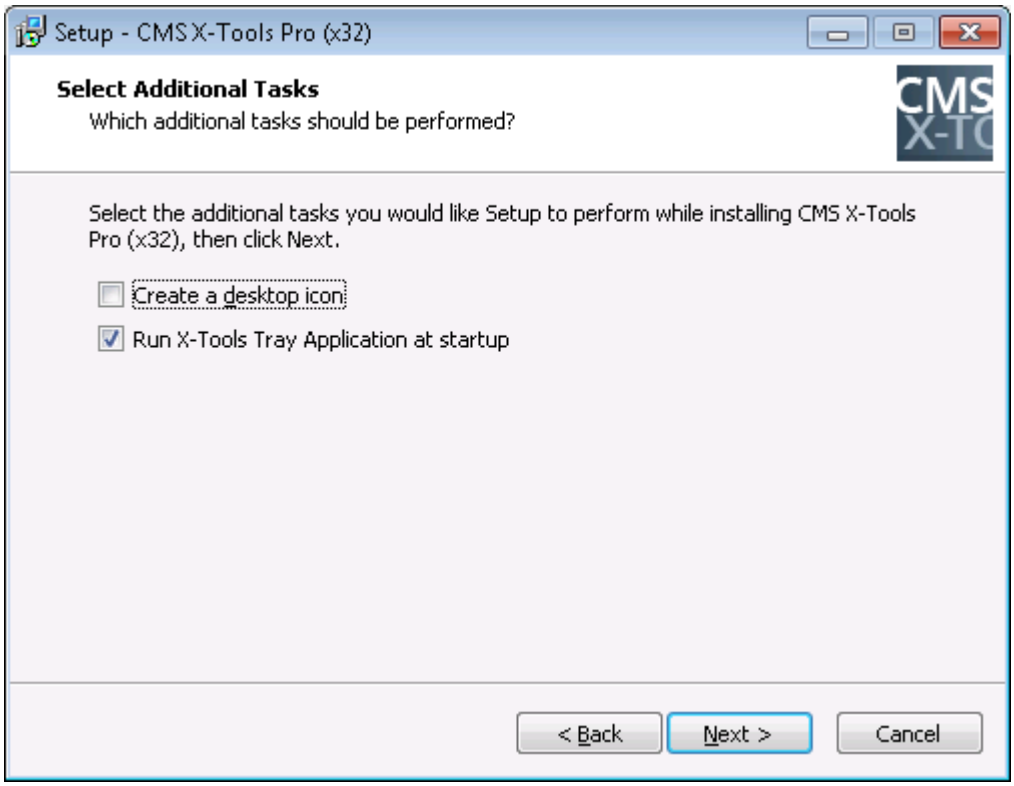

Setup - CMS X-Tools Pro (x32)
Select Additional Tasks
Which additional tasks should be performed?
Select the additional tasks you would like Setup to perform while installing CMS X-Tools Pro (x32), then click Next.
Create a desktop icon
Run X-Tools Tray Application at startup
< Back
Next >
Cancel

*Figure 8: Selection of the additional Tasks which shall be performed by the **Setup***

Before the actual installation starts, ***Setup*** provides an overview about its installation tasks:

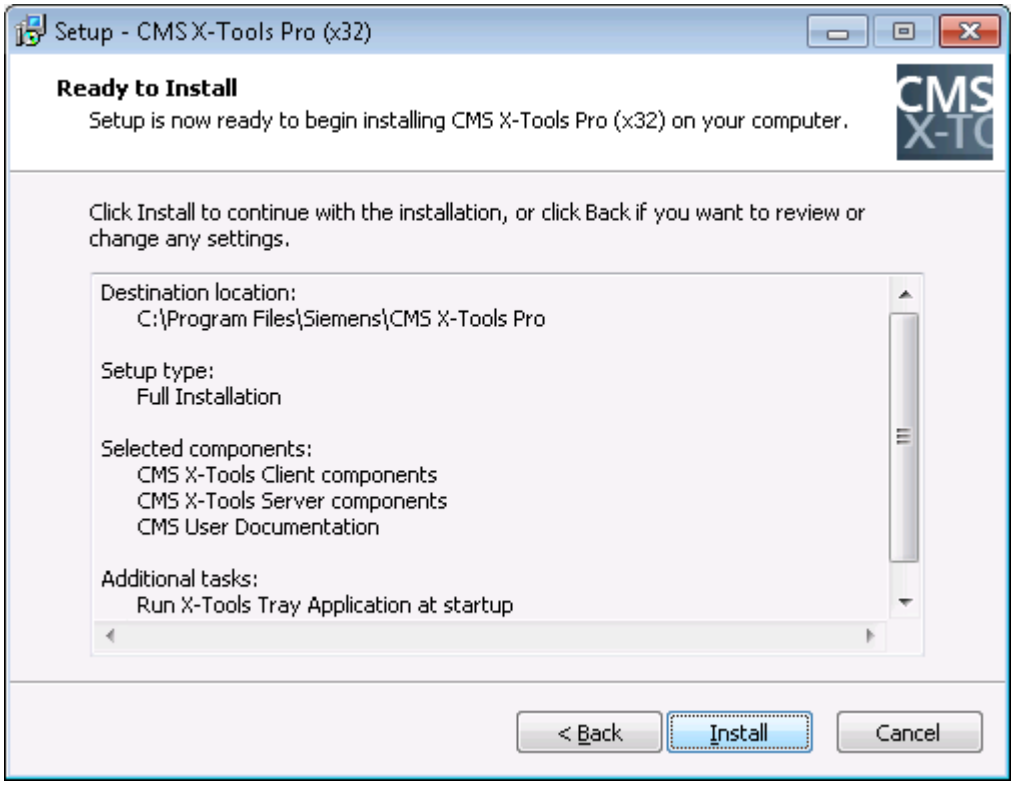

Setup - CMS X-Tools Pro (x32)
Ready to Install
Setup is now ready to begin installing CMS X-Tools Pro (x32) on your computer.
Click Install to continue with the installation, or click Back if you want to review or change any settings.
Destination location:
C:\Program Files\Siemens\CMS X-Tools Pro
Setup type:
Full Installation
Selected components:
CMS X-Tools Client components
CMS X-Tools Server components
CMS User Documentation
Additional tasks:
Run X-Tools Tray Application at startup
< Back
Install
Cancel

Figure 9: Overview about the Installation Tasks

While the installation is in progress, ***Setup*** shows the currently processed operation:

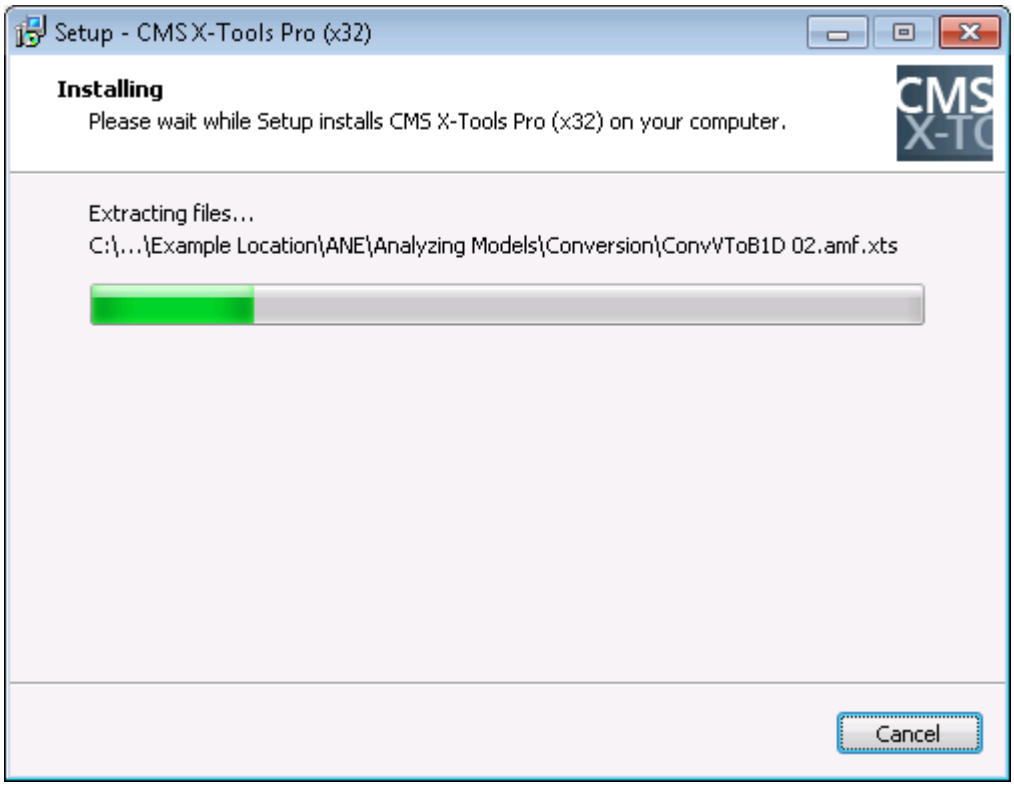

Setup - CMS X-Tools Pro (x32)
Installing
Please wait while Setup installs CMS X-Tools Pro (x32) on your computer.
Extracting files...
C:\...\Example Location\ANE\Analyzing Models\Conversion\ConvVToB1D 02.amf.xts
Cancel

Figure 10: Overview about the Progress of the Installation

At the end of the installation, ***Setup*** shows the according information and allows to start the ***X-Tools Client*** directly:

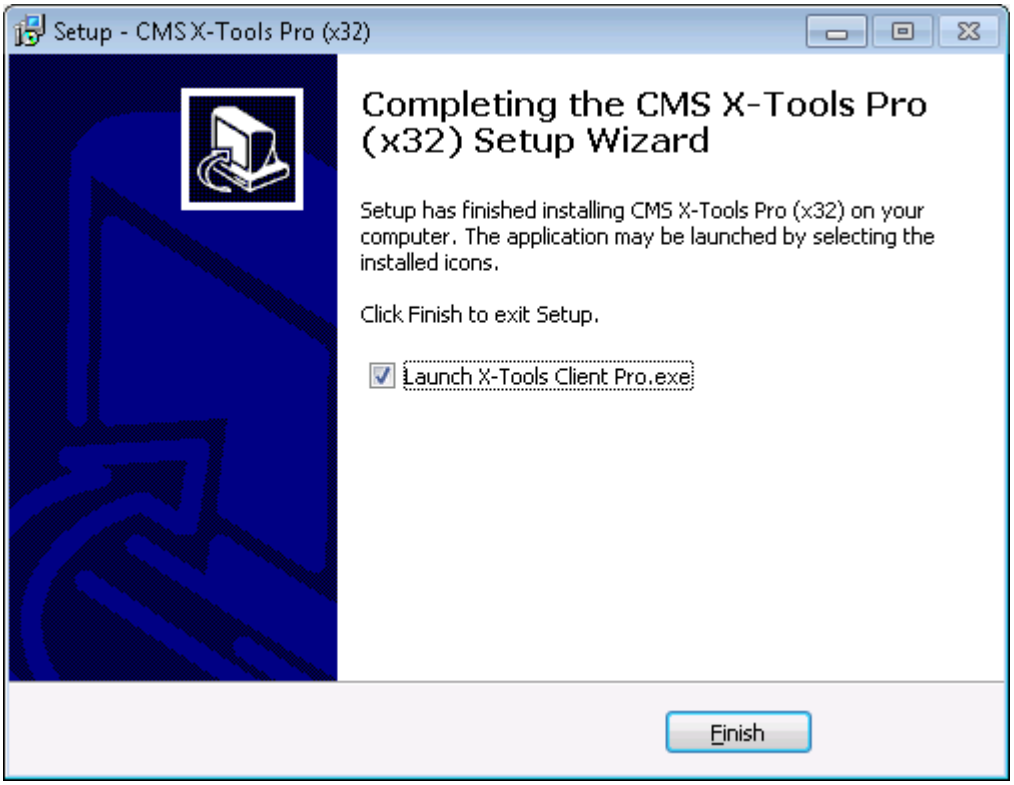

Setup - CMS X-Tools Pro (x32)
Completing the CMS X-Tools Pro (x32) Setup Wizard
Setup has finished installing CMS X-Tools Pro (x32) on your computer. The application may be launched by selecting the installed icons.
Click Finish to exit Setup.
Launch X-Tools Client Pro.exe
Finish

Figure 11: Installation completed

During the installation, ***Setup*** checks if the required software modules for ***X-Tools*** are already installed on the local computer. In case a requirement is missing, ***Setup*** offers to install it.

Missing requirements could be:

- Siemens Automation License Manager
- Microsoft Visual C++ Redistributables
- Microsoft DirectX
- Microsoft .NET Framework

The according additional setups are started automatically in case it is needed. Follow the setup wizards of each required component in order to install it.

The installation of ***X-Tools*** also can be performed without the ***Setup*** by a manual copy of the “Client”, “Server” and/or “Service” directories from the installation media to the target directory. In this case, the other software components like the redistributable files must be installed manually.

4.1.2 Create Shortcuts on the Desktop

The ***Setup*** creates a shortcut to the ***X-Tools Client*** at the desktop automatically. The following paths can be used in order to create additional shortcuts manually:

- Open the Windows Explorer.
- Drag the following file from your installation directory with the left mouse button to the desktop while keeping the <Alt> button pressed:
  ..\CMS X-Tools Pro\Client\System\Bin\X-Tools Client Pro.exe
- Drag the following file from your installation directory with the left mouse button to the desktop while keeping the <Alt> button pressed:
  ..\CMS X-Tools Pro\Documents\CMS X-Tools - User Manual - 01 - Introduction.EN.pdf

4.1.3 Migration from earlier Versions of X-Tools

4.1.3.1 Overview

In general, multiple versions of ***X-Tools*** can be installed simultaneously at the same computer when each version is put into another directory. However, it is not possible to have two instances of the ***X-Tools Server*** running at the same time.

Via the concept of locations for Configuration Files, multiple versions of ***X-Tools*** can use the same Configuration Files so that all changes within one version also become available to the other installed versions. See point 6.5 for further information about locations for Configuration Files.

It must be considered that file compatibility is provided only from earlier to newer versions of ***X-Tools***. Therefore, newer versions of ***X-Tools*** are able to read the files which have been saved by earlier versions, but earlier versions of ***X-Tools*** may not be able to read the files which were saved by newer version.

Before the installation of a newer version of ***X-Tools*** is being started it must be decided whether the previous installation shall be kept or not. The following chapters provide the steps for both kinds of installation, with and without keeping of the previous version.

Independently to this decision, it is recommended to create a backup copy of all relevant files (***X-Tools*** installation folder plus eventually present locations outside the installation directory) before the installation of the new version.

In order to restore the backup of the previous version, empty the installation directory first and copy your backup files into it afterwards. In case there were locations outside the installation directory, empty their directories also and copy the files from your backup afterwards. Emptying of the target directories is important in order to be sure that no files from temporary installations or from any other operations are left and to be sure that the original state of the installation is restored.

4.1.3.2 New IEEE1394 Driver in V 03.03

Together with ***X-Tools*** V 03.03, a new version of the IEEE1394 driver is being delivered. ***X-Tools*** V 03.03 is not compatible with the IEEE1394 driver of previous versions, therefore the new IEEE1394 driver must be installed for all IEEE1394 devices which shall be used together with V 03.03 of ***X-Tools***. The following steps must be performed in order to upgrad the driver of all SIPLUS CMS4000 IEEE1394 devices:

- Close ***X-Tools***.
- Execute “Driver Config (Upgrade).bat” from your “..\Drivers\IEEE1394” directory and wait until the upcoming command window outputs that the upgrade has been completed.
- After the upgrade has finished with success, all SIPLUS CMS4000 IEEE1394 devices will appear below the “PionsysDev” group within the Device Manager of windows.
- The upgrade is being performed only for the IEEE1394 devices which are currently attached to the PC. A manual installation of the new IEEE1394 driver may be necessary for all IEEE1394 devices which are being attached to the PC after the upgrade.
- While the upgrade is in progress, the hardware assistant of Windows may pop up for each device. Typically, it disappears automatically after a few moments while the upgrade is in progress. Depending to the configuration of the operating system it can happen that the automatic installation of the device driver must be chosen within the hardware assistant in order to continue the installation.

Once the new IEEE1394 driver has been installed for a device, the device can not be used with versions of ***X-Tools*** before V 03.03 any more. In order to use the device with versions of ***X-Tools*** before V 03.03 again, the device driver must be set back to the old driver. The following steps must be performed in order to downgrade the IEEE1394 driver of all SIPLUS CMS4000 IEEE1394 devices:

- Close ***X-Tools***.
- Execute “Driver Config (Downgrade).bat” from your “..\Drivers\IEEE1394” directory and wait until the upcoming command window outputs that the downgrade has been completed.
- After the downgrade has finished with success, all SIPLUS CMS4000 IEEE1394 devices will appear below the “SIPLUS CMS4000” group within the Device Manager of windows.

- The downgrade is being performed only for the IEEE1394 devices which are currently attached to the PC. A manual installation of the old IEEE1394 driver may be necessary for all IEEE1394 devices which are being attached to the PC after the downgrade.
- While the downgrade is in progress, the hardware assistant of Windows may pop up for each device. Typically, it disappears automatically after a few moments while the downgrade is in progress. Depending to the configuration of the operating system it can happen that the automatic installation of the device driver must be chosen within the hardware assistant in order to continue the installation. In case Windows asks whether the newer driver file shall be replaced by the older driver file, confirm this operation.

4.1.3.3 Update of Analyzing Models

The interface of Analyzing Functions can change from one version of ***X-Tools*** to the next version. Before the start of each Analyzing Model, the ***X-Tools Server*** verifies whether the interface description of the (Analyzing and User) functions from the Analyzing Model match the interface definitions of the currently present functions. In case the interfaces do not match, the Analyzing Model suspends with an according log entry.

In order to receive a runnable Analyzing Model again, the Analyzing Model must be opened within an **Analyzing Model Editor**. The **Analyzing Model Editor** is able to automatically update the Analyzing Model in most cases. After the Analyzing Model has been updated and saved, it can be started again.

In rare cases it can happen that the Analyzing Model can not be updated automatically. In such a situation the **Analyzing Model Editor** provides detailed information about the functions which could not be updated and the update can be performed manually then.

4.1.3.4 Update from V 03.01 SP1 or earlier to V 04.02

4.1.3.4.1 Parallel Installation to the old Version

With V 03.02 of ***X-Tools***, the internal directory structure has been changed. Therefore, overwriting of the previous installation of ***X-Tools*** can not be performed fully automatically. The following steps must be performed in order to perform the update:

- Create the backup of your existing installation of ***X-Tools*** (see point 4.1.3.1).
- Remove the write protection from all files of the installation directory:
  - Call the context menu for the installation directory within the Windows Explorer and select “Properties”.
  - Deselect the “Read-only” attribute and press “Apply”.
  - In the upcoming dialog, choose “Apply changes to this folder, subfolders and files” and press “OK”.
- Start the installation of the new version of ***X-Tools*** by launching of “Setup.exe” from the installation media.
  - Within the setup, do not specify the current installation directory of ***X-Tools*** as target directory for the ongoing installation. Instead, use another, empty directory as target directory.
- After the setup has been completed, the new version of ***X-Tools*** has been installed.
- Start the new installation of ***X-Tools***, wait until everything has been initialized and close ***X-Tools*** afterwards. During this first start, ***X-Tools*** prepares the basic directory structures which are needed for the following steps.

- Copy the following files from the earlier installation of ***X-Tools*** to the corresponding directories of the new installation (overwrite all files which exist in the target directory already):
  - ..\Client\User\Basic Settings\Global Options.gof.xtc
  - ..\Client\User\Basic Settings\Global Settings.gsf.xtc
  - ..\Client\User\Basic Settings\Layout.lyt.xtc
  - ..\Client\User\Default Location*.*
  - ..\Server\User\Basic Settings\Data Categories.dcf.xts
  - ..\Server\User\Basic Settings\Global Options.gof.xts
  - ..\Server\User\Basic Settings\User Accounts.uaf.xts
  - ..\Server\User\Default Location*.*
  - ..\Server\User\Offline Files*.*
- Be sure that the following files are not being copied - they contain absolute path information and therefore they won't work within the changed (new) directory:
  - ..\Client\User\Basic Settings\File Locations.flf.xtc
  - ..\Server\User\Basic Settings\File Locations.flf.xts
- In case additional locations for Configuration Files or Offline Data where specified within the earlier installation of ***X-Tools***, the same locations also must be added manually to the new installation. See point 6.5 for further information about locations for Configuration Files.
- The new version can be started now and uses all of the Configuration Files and settings from the previous installation.
- In case the previous version is not needed any more, its installation folder can be deleted. Do not delete any of the directories outside the installation directory (namely the locations of Configuration Files and offline data), as they may be used by the new installation.

4.1.3.5 Update from V 03.02 or later to V 04.02

4.1.3.5.1 Overwriting of the old Version

In order to overwrite a V 03.02 or later with the new V 04.02, the following steps must be performed:

- Create the backup of your existing installation of ***X-Tools*** (see point 4.1.3.1).
- Remove the write protection from all files of the installation directory:
  - Call the context menu for the installation directory within the Windows Explorer and select "Properties".
  - Deselect the "Read-only" attribute and press "Apply".
  - In the upcoming dialog, choose "Apply changes to this folder, subfolders and files" and press "OK".
- Start the installation of the new version of ***X-Tools*** by launching of "Setup.exe" from the installation media.
  - Within the setup, specify the current installation directory of ***X-Tools*** as target directory for the ongoing installation.
- After the setup has been completed, the new version of ***X-Tools*** has been installed over the old installation. The new version can be started now and automatically uses all of the Configuration Files and settings from the previous installation.

4.1.3.5.2 Parallel Installation to the old Version

The installation of V 04.02 in parallel to V 03.02 or later can be performed like it is described by point 4.1.3.4.1 for V 03.01 SP1 and earlier.

Although multiple versions of ***X-Tools*** can be installed in parallel, the IEEE1394 driver for the to-be-used SIPLUS CMS4000 IEEE1394 drivers must be upgraded (when V 03.03 or later shall be used) or downgraded (when V 03.02 SP1 or earlier shall be used), as described by point 4.1.3.2.

4.1.4 Updates

In case there are bugfixes needed between major releases of ***X-Tools***, according updates are provided. These updates come with their own installer and are able to update the existing installation of ***X-Tools*** automatically.

4.2 Hardware Installation

4.2.1 How to connect CMS to your Application

CMS is designed to work as far as possible as a reactionless system to the existing process and automation system. Because of this attempt, binary and analog signals which are available anywhere in the process can be connected to the high resistance input ports of the ***IONs*** and/or ***IFNs*** in parallel to the existing system.

To extract data from bus systems like PROFIBUS DP, the available ***IONs*** have to be added simply as physical bus members. There are no changes in the configuration of these buses needed because the according ***IONs*** are realized as spy modules and are not logical members of the bus.

Different software ***IONs*** are available for several SIMATIC automation controllers. These software ***IONs*** can be added to the application program of the automation controller and can be used in order to transmit any of the internal variables of the controller to ***X-Tools***.

4.2.2 Communication Interfaces

IEEE1394 ***IONs*** and ***IFNs*** communicate with ***X-Tools*** via IEEE1394. Examples for IEEE1394 ***IONs*** and ***IFNs*** are:

- IFN ANALOGINPUT
- IFN ANALOGINPUT-DISTANCE
- IFN VIB-ACC
- ION AnalogInput T001
- ION BinaryInput T001
- ION PROFIBUS DP Spy T001
- ION VIB-A T001
- ION VIB-D T001

Ethernet ***IONs*** communicate with ***X-Tools*** via Ethernet. Examples for Ethernet ***IONs*** are:

- CMS2000
- ION SIMATIC S7-1200
- ION SIMATIC S7-1500
- ION SIMATIC S7-300 T001
- ION SIMATIC S7-400 T001
- ION SIMATIC S7 PN
- ION SIMATIC TDC T001
- ION SIMOTION T001

Serial port ***CTNs*** communicate with ***X-Tools*** via the serial port. Examples for serial port ***CTNs*** are:

- CTN WatchGuard T001

OPC UA servers can be accessed by ***X-Tools*** in its role as OPC UA client. OPC UA servers appear within ***X-Tools*** as devices of type

- OPC UA T001

X-Tools can act as OPC UA server. In this role, it can be accessed by other OPC UA client applications in order to read the current online data values from ***X-Tools***.

4.2.3 How to connect IEEE1394 Devices to the PC

IEEE1394 Cable

In order to connect an IEEE1394 ***ION*** or ***IFN*** to ***X-Tools***, an IEEE1394 cable must be plugged between the used PC and the ***ION*** or ***IFN***. To connect further ***IONs*** or ***IFNs***, connect them via an IEEE1394 cable to any other, already connected ***ION*** or ***IFN*** or to another free IEEE1394 port at the PC.

Note

IEEE1394 allows any topology (e.g. chain, tree) except a loop. Up to 16 IEEE1394 devices can be connected in a chain. In case more than 16 IEEE1394 devices have to be connected together, a tree topology has to be created.

Note

Do not use multiple IEEE1394 host adapters simultaneously for connecting of IEEE1394 devices to a single PC. Configurations with multiple used IEEE1394 host adapters are not supported by the system.

Note

Microsoft Windows 7 is delivered with two versions of the basic IEEE1394 driver. The so-called “new” driver does not work together with all kinds of IEEE1394 host adapters because it does not support IEEE1394-1995 PHYs. In case the IEEE1394 devices are not recognized properly by Microsoft Windows 7, consider to switch to the so-called “legacy” driver. See also point 7.1.4.

Power Supply

To use an ***ION*** or ***IFN*** you have to provide it with a 24 V DC power supply.

Driver Installation

After connecting an IEEE1394 ***ION*** or ***IFN*** to its power supply and to the PC via IEEE1394 for the first time, the Windows operating system automatically will detect a new hardware device. To use it, the corresponding SIPLUS CMS4000 driver must be installed for it.

Whenever an IEEE1394 device is being attached to the PC for the first time, the hardware wizard of Windows starts automatically and wants to install a driver for the new device. For the installation, the following sub-path of your installation directory has to be provided to the hardware wizard of Windows:

..\CMS X-Tools Pro\Drivers\IEEE1394\

IEEE1394 ***IONs*** and ***IFNs*** are detected by ***X-Tools*** automatically after the driver installation and can be used by ***X-Tools*** immediately.

4.2.4 How to connect Ethernet IONs to the PC

Ethernet Cable

In order to connect an Ethernet ***ION*** to ***X-Tools***, an Ethernet cable must be plugged between the used PC and the ***ION***. Ethernet standard networking devices like switches and/or hubs can be used between the PC and the ***ION*** as well.

Power Supply

The CMS part of current Ethernet ***IONs*** is limited to a piece of software which is installed on a corresponding hardware. Therefore, the hardware of Ethernet ***IONs*** is non-CMS and has its own power supply instructions.

Driver Installation

There is no driver installation needed for Ethernet ***IONs***. Instead, Ethernet ***IONs*** have to be announced manually to ***X-Tools*** after they have been attached and before they can be used. The **Interface Profile Editor** for the socket interface (see **IPE Socket T001**) must be used for the configuration of connections to Ethernet ***IONs.***

4.2.5 How to connect USB CTNs to the PC

USB Cable

In order to connect a USB ***CTN*** to the used PC, a USB cable must be plugged between the used PC and the ***CTN***.

Power Supply

USB ***CTNs*** do not need any external power supply because they are powered via the USB cable from the used PC.

Driver Installation

After connecting a USB ***CTN*** to the PC via USB for the first time, the Windows operating system automatically will detect a new hardware device. To use it, the corresponding driver must be installed for it.

The used driver installs a virtual COM port, so that the ***CTN*** becomes visible below the “Ports (COM and LPT)” item of the Windows Device Manager with its COM number. As a result, USB ***CTNs*** are accessed like serial devices by ***X-Tools***.

Whenever a USB ***CTN*** is being attached to the PC for the first time, the hardware wizard of Windows starts automatically and wants to install a driver for the new device. For the installation, the following path has to be provided to the hardware wizard of Windows:

..\CMS X-Tools Pro\Drivers\USB\

USB ***CTNs*** have to be announced manually to ***X-Tools*** after they have been attached and before they can be used. They have to be announced as serial devices because the installed USB driver emulates a virtual COM port which must be used by ***X-Tools*** in order to be able to access the device. The **Interface Profile Editor** for the serial port interface (see **IPE Serial Port T001**) must be used for the configuration of connections to USB ***CTNs***.

4.2.6 How to connect OPC UA Servers to the PC

Ethernet Cable

In order to connect an OPC UA server to ***X-Tools***, an Ethernet cable must be plugged between the PC which is running ***X-Tools*** and the device which is running the OPC UA server. In case the OPC UA server is running at the same PC as ***X-Tools***, no additional Ethernet cable is needed.

Power Supply

The device which is providing the OPC UA server must be powered on as specified by the manufacturer of the device. In case the OPC UA server is running at the same PC as ***X-Tools***, no additional power supply is needed.

Driver Installation

There is no driver installation needed for OPC UA servers. Instead, OPC UA servers have to be announced manually to ***X-Tools*** after they have been attached and before they can be used. The **Interface Profile Editor** for the OPC UA interface (see **IPE OPC UA T001**) must be used for the configuration of connections to OPC UA servers.

4.2.7 How to connect OPC UA Clients to the PC

Ethernet Cable

In order to connect an OPC UA client to ***X-Tools***, an Ethernet cable must be plugged between the PC which is running ***X-Tools*** and the device which is running the OPC UA client. In case the OPC UA client is running at the same PC as ***X-Tools***, no additional Ethernet cable is needed.

Power Supply

The device which is acting as OPC UA client must be powered on as specified by the manufacturer of the device. In case the OPC UA client is running at the same PC as ***X-Tools***, no additional power supply is needed.

Driver Installation

There is no driver installation needed for OPC UA client. From the point of view of ***X-Tools***, the only needed configuration for the communication with OPC UA clients is to enable the OPC UA server functionality of ***X-Tools***.

4.2.8 Video Card Driver

It is highly recommended to ensure that the used video card driver is up to date in order to achieve the best possible performance for the visualization of data within the **Monitoring System**. As the **Monitoring System** allows choosing between DirectX and OpenGL renderers for the visualization, a good support of the chosen renderer (DirectX or OpenGL) is highly recommended for fluent visualization results.

4.2.9 Minimal Firmware Versions

The following minimal firmware versions must be provided by each device:

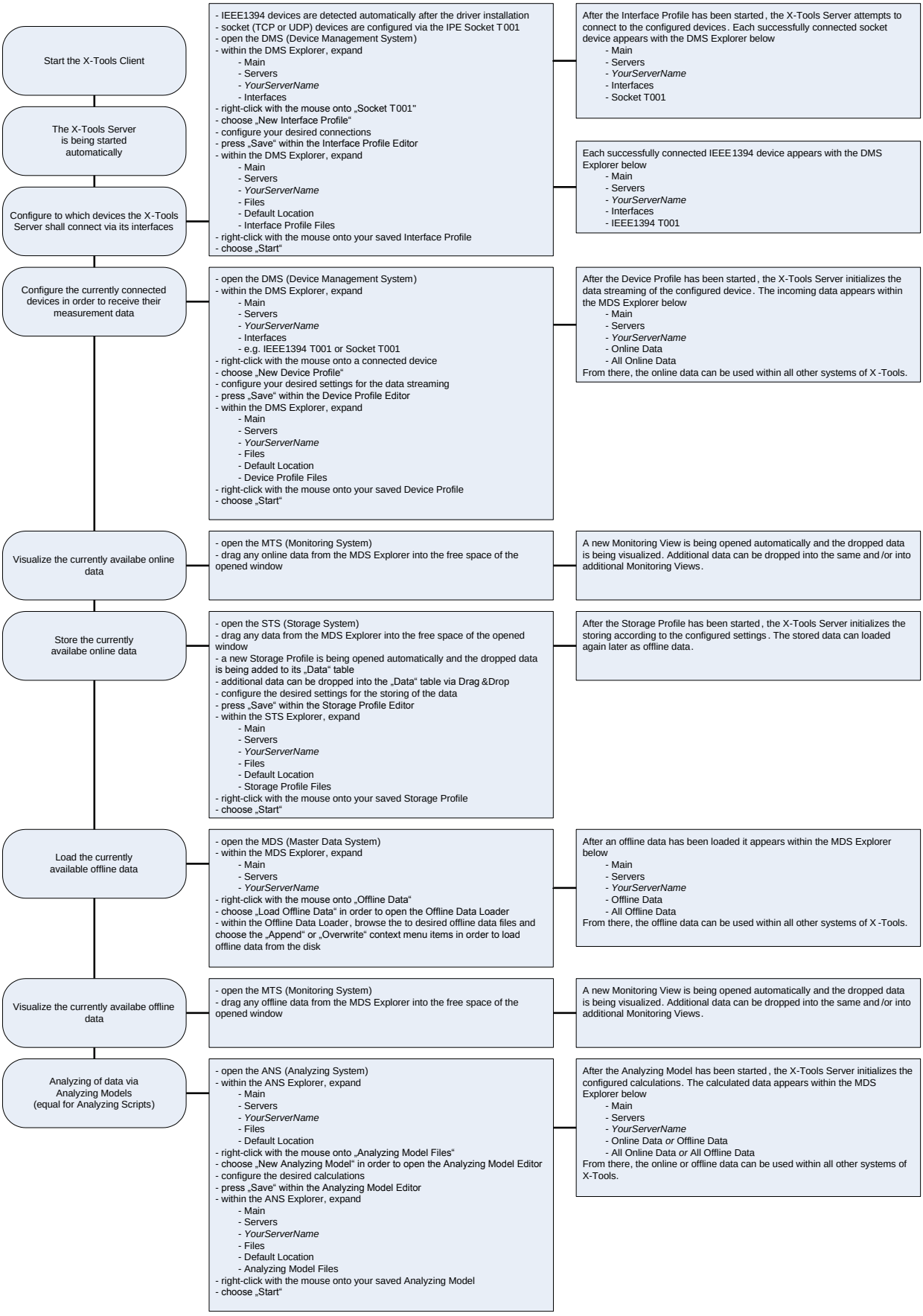
Device	Minimal CMM/Main Version	Minimal IOM/IFM Version
CTN WatchGuard T001	V 02.05.EN.0006	n/a
CMS2000	V3.0	n/a
IFN ANALOGINPUT	V 01.12.EN.0000	V 01.02.EN.0004
IFN ANALOGINPUT-DISTANCE	V 01.12.EN.0000	V 01.04.EN.0000
IFN VIB-ACC	V 01.12.EN.0000	V 01.03.EN.0018
ION SIMATIC S7 PN	V 01.02.EN.0000	n/a
ION SIMATIC S7-1200	V 01.01	n/a
ION SIMATIC S7-1500	V 01.01	n/a
ION SIMATIC S7-300 T001	V 03.01.EN.0000	n/a
ION SIMATIC S7-400 T001	V 03.06.EN.0000	n/a
ION SIMATIC TDC T001	V 03.03.EN.0004	n/a
ION SIMOTION T001	V 01.02.EN.0000	n/a
ION AnalogInput T001	V 03.00.EN.0010	V 08.00.EN.0000
ION BinaryInput T001	V 03.00.EN.0010	V 02.00.EN.0001
ION PROFIBUS DP Spy T001	V 03.00.EN.0010	n/a
ION VIB-A T001	V 03.00.EN.0010	V 04.00.EN.0000
ION VIB-D T001	V 03.00.EN.0010	V 04.00.EN.0000

In case your device reports an older firmware version it must be updated to the latest available version. Detailed information about how to update the firmware of devices is obtained from your local Siemens office and from the homepage http://www.siemens.com/siplus-cms.

Note

Devices with an older firmware version than above may also work properly. However, older firmware versions are not tested any more together with the latest version of ***X-Tools***, thus it is always suggested to use the latest available version of all device firmware together with the current version of ***X-Tools***.

5 Quick Reference


Start the X-Tools Client
The X-Tools Server is being started automatically
Configure to which devices the X-Tools Server shall connect via its interfaces
- IEEE1394 devices are detected automatically after the driver installation
- socket (TCP or UDP) devices are configured via the IPE Socket T001
- open the DMS (Device Management System)
- within the DMS Explorer, expand
- Main
- Servers
- YourServerName
- Interfaces
- right-click with the mouse onto „Socket T001"
- choose „New Interface Profile"
- configure your desired connections
- press „Save" within the Interface Profile Editor
- within the DMS Explorer, expand
- Main
- Servers
- YourServerName
- Files
- Default Location
- Interface Profile Files
- right-click with the mouse onto your saved Interface Profile
- choose „Start"
After the Interface Profile has been started, the X-Tools Server attempts to connect to the configured devices. Each successfully connected socket device appears with the DMS Explorer below
- Main
- Servers
- YourServerName
- Interfaces
- Socket T001
Each successfully connected IEEE1394 device appears with the DMS Explorer below
- Main
- Servers
- YourServerName
- Interfaces
- IEEE1394 T001
Configure the currently connected devices in order to receive their measurement data
- open the DMS (Device Management System)
- within the DMS Explorer, expand
- Main
- Servers
- YourServerName
- Interfaces
- e.g. IEEE1394 T001 or Socket T001
- right-click with the mouse onto a connected device
- choose „New Device Profile"
- configure your desired settings for the data streaming
- press „Save" within the Device Profile Editor
- within the DMS Explorer, expand
- Main
- Servers
- YourServerName
- Files
- Default Location
- Device Profile Files
- right-click with the mouse onto your saved Device Profile
- choose „Start"
After the Device Profile has been started, the X-Tools Server initializes the data streaming of the configured device. The incoming data appears within the MDS Explorer below
- Main
- Servers
- YourServerName
- Online Data
- All Online Data
From there, the online data can be used within all other systems of X-Tools.
Visualize the currently availabe online data
- open the MTS (Monitoring System)
- drag any online data from the MDS Explorer into the free space of the opened window
A new Monitoring View is being opened automatically and the dropped data is being visualized. Additional data can be dropped into the same and/or into additional Monitoring Views.
Store the currently availabe online data
- open the STS (Storage System)
- drag any data from the MDS Explorer into the free space of the opened window
- a new Storage Profile is being opened automatically and the dropped data is being added to its „Data" table
- additional data can be dropped into the „Data" table via Drag &Drop
- configure the desired settings for the storing of the data
- press „Save" within the Storage Profile Editor
- within the STS Explorer, expand
- Main
- Servers
- YourServerName
- Files
- Default Location
- Storage Profile Files
- right-click with the mouse onto your saved Storage Profile
- choose „Start"
After the Storage Profile has been started, the X-Tools Server initializes the storing according to the configured settings. The stored data can loaded again later as offline data.
Load the currently available offline data
- open the MDS (Master Data System)
- within the MDS Explorer, expand
- Main
- Servers
- YourServerName
- right-click with the mouse onto „Offline Data"
- choose „Load Offline Data" in order to open the Offline Data Loader
- within the Offline Data Loader, browse the to desired offline data files and choose the „Append" or „Overwrite" context menu items in order to load offline data from the disk
After an offline data has been loaded it appears within the MDS Explorer below
- Main
- Servers
- YourServerName
- Offline Data
- All Offline Data
From there, the offline data can be used within all other systems of X-Tools.
Visualize the currently availabe offline data
- open the MTS (Monitoring System)
- drag any offline data from the MDS Explorer into the free space of the opened window
A new Monitoring View is being opened automatically and the dropped data is being visualized. Additional data can be dropped into the same and/or into additional Monitoring Views.
Analyzing of data via Analyzing Models (equal for Analyzing Scripts)
- open the ANS (Analyzing System)
- within the ANS Explorer, expand
- Main
- Servers
- YourServerName
- Files
- Default Location
- right-click with the mouse onto „Analyzing Model Files"
- choose „New Analyzing Model" in order to open the Analyzing Model Editor
- configure the desired calculations
- press „Save" within the Analyzing Model Editor
- within the ANS Explorer, expand
- Main
- Servers
- YourServerName
- Files
- Default Location
- Analyzing Model Files
- right-click with the mouse onto your saved Analyzing Model
- choose „Start"
After the Analyzing Model has been started, the X-Tools Server initializes the configured calculations. The calculated data appears within the MDS Explorer below
- Main
- Servers
- YourServerName
- Online Data or Offline Data
- All Online Data or All Offline Data
From there, the online or offline data can be used within all other systems of X-Tools.

6 Tutorials

6.1 General Description

The following tutorials provide a fast and detailed overview about the key functionalities of ***X-Tools***. They build on each other, therefore it is strongly recommended to take a look onto them in the order in which they are provided in case of the first contact to the system.

However, in case you already have some experience with the system, you may find the one or the other tutorial useful in order to recall a certain functionality or handling.

In addition to this tutorial, examples for all available configurations are delivered together with ***X-Tools***. These examples are being found in a separated Configuration File location so that they can be enabled or disabled with only a few clicks at any time. Take a look at point 6.5 in order to see how to enable or disable additional Configuration File locations.

6.2 X-Tools Basics

6.2.1 General Description

In this chapter you will learn more about the basic modules and functionalities of ***X-Tools***.

This chapter contains the following topics:

- **Running X-Tools for the first Time**
- **X-Tools User Interface**
- **Available Data**

6.2.2 Running X-Tools for the first Time

Start X-Tools

- To start ***X-Tools***, double click the ***X-Tools*** shortcut on your desktop. Alternatively, you can directly call "X-Tools Client Pro.exe" from the following directory:
  "..\CMS X-Tools Pro\Client\System\Bin"

System Paths

- At the first startup, ***X-Tools*** automatically generates the default configuration of all settings, options and paths. By default, the started ***X-Tools Client*** automatically also starts the ***X-Tools Server*** and establishes the connection so that no additional configuration is needed and the actual measurement tasks can be started immediately.
- Via the **Global Options Editor**, you are able to modify any of the default settings, options and paths of the ***X-Tools Client*** and of the ***X-Tools Server*** at any later time. Take a look at point 6.4.2 in order to receive detailed information about the **Global Options Editor** and its functionalities.

Ready

At this point, the basic configuration of ***X-Tools*** has been finished and ***X-Tools*** is ready to use!

6.2.3 X-Tools User Interface

Sub Systems

The user interface of the ***X-Tools Client*** is divided into the following six sub systems:

- **MDX** - **Master Data System Explorer**
- **MMS** - **Main Management System**
- **DMS** - **Device Management System**
- **MTS** - **Monitoring System**
- **ANS** - **Analyzing System**
- **STS** - **Storage System**

Main Title Bar

The main title bar of the ***X-Tools Client*** contains the following functionalities (left to right):

- Caption
- Show/Hide Master Data System Explorer button - shows/hides the **Master Data System Explorer**
- Full Screen button - puts the ***X-Tools Client*** into full screen mode
- Minimize button - minimizes the ***X-Tools Client***
- Maximize or Restore button - maximizes or restores the ***X-Tools Client***
- Exit button - closes the ***X-Tools Client***

*Figure 12: Main Title Bar of the **X-Tools Client***

Main Menu Bar

The main menu bar of the ***X-Tools Client*** contains the following functionalities (left to right):

- MDS Explorer button - shows/hides the **Master Data System Explorer**
- MMS button - brings the **Main Management System** to the front in case it is opened already and opens it in case it is not opened yet
- DMS button - brings the **Device Management System** to the front in case it is opened already and opens it in case it is not opened yet
- MTS button - brings the **Monitoring System** to the front in case it is opened already and opens it in case it is not opened yet
- ANS button - brings the **Analyzing System** to the front in case it is opened already and opens it in case it is not opened yet
- STS button - brings the **Storage System** to the front in case it is opened already and opens it in case it is not opened yet

*Figure 13: Main Menu Bar of the **X-Tools Client***

Main Status Bar

The main status bar of the ***X-Tools Client*** contains the following functionalities (left to right):

- visualization of the current status message, in case there is any
- combo box for the selection of the ***X-Tools Server*** in case one or more are available, otherwise the current status message is displayed

- icons which represent the current status of the Configuration Files of the currently chosen ***X-Tools Server*** via the icon itself (e.g. through colors) and via the tool tip of each icon:
  - status of Interface Profiles
  - status of Device Profiles
  - status of Analyzing Models
  - status of Analyzing Scripts
  - status of Storage Profiles
- current date and time of the ***X-Tools Client***

*Figure 14: Main Status Bar of the **X-Tools Client***

Window Size and Position

The following window size and position functionalities are provided by the ***X-Tools Client***:

- The Full Screen button from the title bar each window can be used in order to put it into full screen mode.

Figure 15: Full Screen Button

- In full screen mode, the Restore button can be used in order to bring the window back to its original size. Alternatively, a double-click onto the title bar of a window which is in full screen mode also brings the window back to its original size.

Figure 16: Restore Button

- The Maximize button can be used in order to maximize a window within its workspace. Alternatively, a double-click onto the title bar of a window which is not maximized also maximizes the window within its workspace.

Figure 17: Maximize Button

- In case a window is maximized within its workspace, the Restore button can be used in order to bring the window back to its original size. Alternatively, a double-click onto the title bar of a window which is maximized within its workspace also brings the window back to its original size.
- The size of each window can be changed via Drag&Drop. In order to change the size of a window, click onto the border or an edge of the window with the left mouse button and keep the mouse button pressed while moving the mouse cursor. During the mouse move, a light frame shows the new size which the window would get when the mouse button is released.
- The position of each window can be moved via Drag&Drop. In order to move a window to another place within its workspace, click onto the title bar of the window with the left mouse button and keep the mouse button pressed while moving the mouse cursor. During the mouse move, a light frame shows where the dragged window would be dropped when the mouse button is released.
- The detailed behavior of each window can be configured via the context menu which is being called via a click onto the title bar of the window with the right mouse button. The provided context menu allows switching the window mode between tabbed, automatic and manual and also provides some further possibilities of configuration.
- The Close button from the title bar of each window can be used in order to close the window.

Figure 18: Close Button

Tool Tips of Menu Buttons

Each button from the menu bar of a dialog provides a tool tip which describes its functionality in short words. In order to display the tool tip of a menu button, simply move the mouse cursor above the menu button and wait some moments until the tool tip is being displayed.

Try it out

➔ After opening of the ***X-Tools Client***, press the DMS and MTS buttons in order to open the **Device Management System** and the **Monitoring System**.

You can see that the DMS and MTS buttons are highlighted. In addition, at the top of the screen, you can see the **Device Management System** in the background and the **Monitoring System** in the foreground.

➔ Within the **MTS Explorer**, expand the “Main -> Editors -> Monitoring Charts” branch and drag the **MTC yt T001** with the left mouse into the empty space at the right side of the **MTS Explorer**.

A new **Monitoring View Editor** which contains the dropped **MTC yt T001** is being opened automatically.

➔ Drag another **MTC yt T001** from the **MTS Explorer** into the already opened **Monitoring View Editor**.

The second **MTC yt T001** is added to the **Monitoring View Editor**, which contains two **MTC yt T001** now.

➔ Use the Full Screen button of one **MTC yt T001** in order to put it into full screen mode. Double-click onto the title bar of the **MTC yt T001** in full screen mode afterwards in order to put it back to its previous size.

➔ Click into the title bar of one **MTC yt T001** and keep the left mouse button pressed while moving the mouse. Place the dragged **MTC yt T001** right to the second **MTC yt T001** within the **Monitoring View Editor**.

➔ Use the Maximize button of one **MTC yt T001** in order to maximize the **MTC yt T001** within the **Monitoring View Editor**.

➔ Move the mouse cursor onto some menu buttons in order to get the tool tips of the menu buttons being displayed.

Ready

At this point, you know the basics of the user interface of the ***X-Tools Client*** and you know how to arrange the opened windows to your needs.

6.2.4 Available Data

Master Data System Explorer

The **Master Data System Explorer** displays all of the data which is currently available at the ***X-Tools Server***. From there, the currently available data can be moved to the other systems of ***X-Tools*** via simple Drag&Drop operations.

Online Data

Online data is defined as the data which is currently received from the attached devices and which is configured via Device Profiles (see point 6.7).

In addition, Analyzing Models and Analyzing Scripts also can be configured to produce output data of class “online” (see point 6.13 and point 6.14).

Online data is maintained by the ***X-Tools Server*** within ring buffers that are overwritten from the beginning whenever the end of the buffer is being reached. The size of the ring buffer of each data can be configured individually, thus the amount of time which is available for each online data in memory can be determined by the user. In order to save some online data permanently, Storage Profiles (see point 6.9) must be used.

Each ring buffer of online data should be configured to a size which can hold at least a few seconds of data. For example, in case a data acquisition is running with 40 kHz, a ring buffer size of 400.000 values would be able to keep 10 seconds of the data. In case of a ring buffer which is too small to hold the values of its data for a few seconds, an according warning is written into the System Log of the ***X-Tools Server***.

Offline Data

Offline data is defined as the data which has been stored to disk previously and which has been loaded via an **Offline Data Loader** (see point 6.10).

In addition, Analyzing Models and Analyzing Scripts also can be configured to produce output data of class “offline” (see point 6.13 and point 6.14).

Offline data is maintained by the ***X-Tools Server*** within static buffers which are modified only in case an according operation is started by the user, e.g. in case additional offline data is appended, in case the offline data is overwritten with other values or in case an Analyzing Model or Analyzing Script is executed again.

Scalar Data

Scalar data can be used in order to store a single value, either as output of some calculation, as input of some calculation or for both. Scalar data can be created and modified via the **MDS Explorer** and by the Analyzing Fucntions OutputScalar() and OutputScalarString(). In addition, the Monitoring Chart **MTC Mimic Board T001** can be used in order to write the value of a scalar data.

The Analyzing Functions InputScalar() and InputScalarString() can be used in order to use a acalar data within a calculation of the Analyzing System (= inside Analyzing Models, Analyzing Scripts, User Functions and within the Formula Editor table of the **MVE Standard T001**).

There can be multiple software modules which write to the same scalar data at virtually the same time. The value (and properties) of the scalar data match the value (and properties) which have been set by the last software module that has attempted to write them.

All scalar data with a “Lifetime” value of “permanent” are stored by the ***X-Tools Server*** to the following file:

- ..\Server\User\Scalar Data\Scalar Data.csv

During each startup of the ***X-Tools Server***, the contents of this .csv file are read and the contained scalar data are restored together with all of their properties, their last modification time and their last value.

At the moment, scalar data can not be stored (via the Storage System and/or via data snapshots), they can not be used as trigger data (of Storage Profiles and/or Monitoring Views) and they can not be used as storage ID data. In addition, only the **MTC Mimic Board T001**, **MTC yn T001** and **MTC yt T001** are able to visualize the current value of scalar data.

6.2.5 Configuration Files

Types

Configuration Files are being used in order to configure all of the tasks of ***X-Tools***. The following types of Configuration Files are being provided by ***X-Tools***:

Configuration File Type	Description
Main Profiles	Main Profiles are the only type of Configuration Files which are being configured for the ***X-Tools Client*** and are being used in order to connect from the ***X-Tools Client*** to one or multiple ***X-Tools Servers***. See point 6.3.
Interface Profiles	Interface Profiles are being used in order to connect the ***X-Tools Server*** to devices, e.g. via socket (UDP, TCP) or serial port interfaces. After devices have been connected, their measurement data can be acquired via Device Profiles. See point 6.6.
Device Profiles	Device Profiles are being used in order to configure the measurement data which shall be delivered by a certain device. See point 6.7
Device Setups	Device Setups are being used in order to store temporary data which is needed for the configuration or execution of Device Profiles, like the address books of ION SIMATIC TDC T001 devices. They are not editable by the user and are written and read automatically by ***X-Tools***.
Monitoring Views	Monitoring Views are being used in order to visualize any of the currently present (online and/or offline) data. In addition, Monitoring Views can be configured to wait for trigger events and to create automatic reports. See point 6.8, 6.11 and 6.27.
Analyzing Models	Analyzing Models are being edited graphically and are being used in order to perform calculations which base on online and/or offline data. See point 6.13.
Analyzing Scripts	Analyzing Scripts are being edited textually and are being used in order to perform calculations which base on online and/or offline data. See point 6.14.
User Functions	User Functions are being edited graphically and are being used in order to encapsulate analysis tasks, in order to make analysis tasks re-usable and in order to protect the know-how onto which the analysis task bases. See point 6.15.
Storage Profiles	Storage Profiles are being used in order to store any of the currently present (online or offline) data. See point 6.9 and 6.16.
Loading Profiles	Loading Profiles are being used in order to pre-configure which offline data shall be loaded out of the possibly massive amount of present offline data. See point 6.10.

Handling

Most Configuration Files can be started, paused, continued and stopped by the user. After a Configuration File has been started, ***X-Tools*** does not care about the actual file at the disk any more because it already has loaded the relevant contents. Therefore, the following conditions apply for all Configuration Files of ***X-Tools***:

- In case a Configuration File is being updated through the user (e.g. via ***X-Tools*** or directly via overwriting of the file outside of ***X-Tools***), the modified contents are not being taken over by X-Tools automatically. The changes are being taken over in the following cases:
  - when the Configuration File is being stopped and restarted (e.g. also after a restart of ***X-Tools***)
  - in case a Configuration File has been suspended and "Automatic Takeover of changed Configuration Files" is set to "Perform automatic Takeover only in case the Configuration File is suspended" within the global options
  - in case "Automatic Takeover of changed Configuration Files" is set to "Perform automatic Takeover whenever the Source File at the Disk has been changed" within the global options
- The actual file at the disk may be renamed, moved or deleted (e.g. via ***X-Tools*** or directly outside of ***X-Tools***), in which case it also disappears from the according explorer tree of ***X-Tools***. In all of these cases the actual execution of the already running Configuration File is not being interrupted, thus the according explorer tree of ***X-Tools*** may not be able to display all of the currently running Configuration Files.
  - At the next start of ***X-Tools*** the missing Configuration File will not be started any more because ***X-Tools*** obviously has no information about its contents.
- All Configuration Files can be edited also in case their source (e.g. an interface, a device or some data) is not available. Thus, the full configuration of ***X-Tools*** can be created and/or edited also without a direct connection to the final application.

6.3 Client to Server Connectivity

6.3.1 General Description

In this chapter you will learn how the ***X-Tools Client*** can connect to one or more ***X-Tools Servers*** via the functionality of Main Profiles. As all data acquisition, calculation and storage tasks are performed by the ***X-Tools Server***, the ***X-Tools Client*** must be connected to at least one ***X-Tools Server*** in order to configure and examine measurement and analysis tasks. The client server architecture of ***X-Tools*** has been chosen in order to fulfill various requests like full remote configuration possibility or execution of the actual data acquisition, calculation and storage tasks without a user interface and as Windows service.

A Main Profile contains all of the information which is needed in order to connect to one or more ***X-Tools Servers***. This information includes the IP address and ports of the target ***X-Tools Servers*** as well as the username and password which shall be used when connecting. From the provided username and password, the ***X-Tools Server*** determines the access rights for the connected ***X-Tools Client***. Multiple Main Profiles can be created for the ***X-Tools Client***, but only one Main Profile can be started at a time. The ***X-Tools Client*** connects to all of the ***X-Tools Servers*** which are specified via the currently started Main Profile.

The ***X-Tools Client*** is delivered with a default Main Profile which automatically is being started during the startup of the ***X-Tools Client*** and establishes a connection to the local ***X-Tools Server***. In case the remote capabilities of ***X-Tools*** are not needed, the provided default Main Profile allows the usage of ***X-Tools*** without the need to bother about any client server specific issues.

This chapter contains the following topics:

- **Creation of new Main Profiles**
- **Opening of existing Main Profiles**
- **Starting and Stopping of Main Profiles**
- **Securing of the X-Tools Server**
- **X-Tools Server Connectivity Dialog**

6.3.2 Creation of new Main Profiles

Open the MMS Explorer

- Open the **Main Management System** and ensure that the **MMS Explorer** is displayed at the left side of the **Main Management System** workspace.

Functionality

- Within the **MMS Explorer**, expand the “Main -> Editors -> Main Profile Editors” branch.
- Drag the **MPE Standard T001** with the left mouse into the empty space at the right side of the **MMS Explorer**. An empty **MPE Standard T001** is opened automatically.
- Press the New button in order to create a new Main Profile. The new Main Profile is displayed by the **MPE Standard T001** with default values.

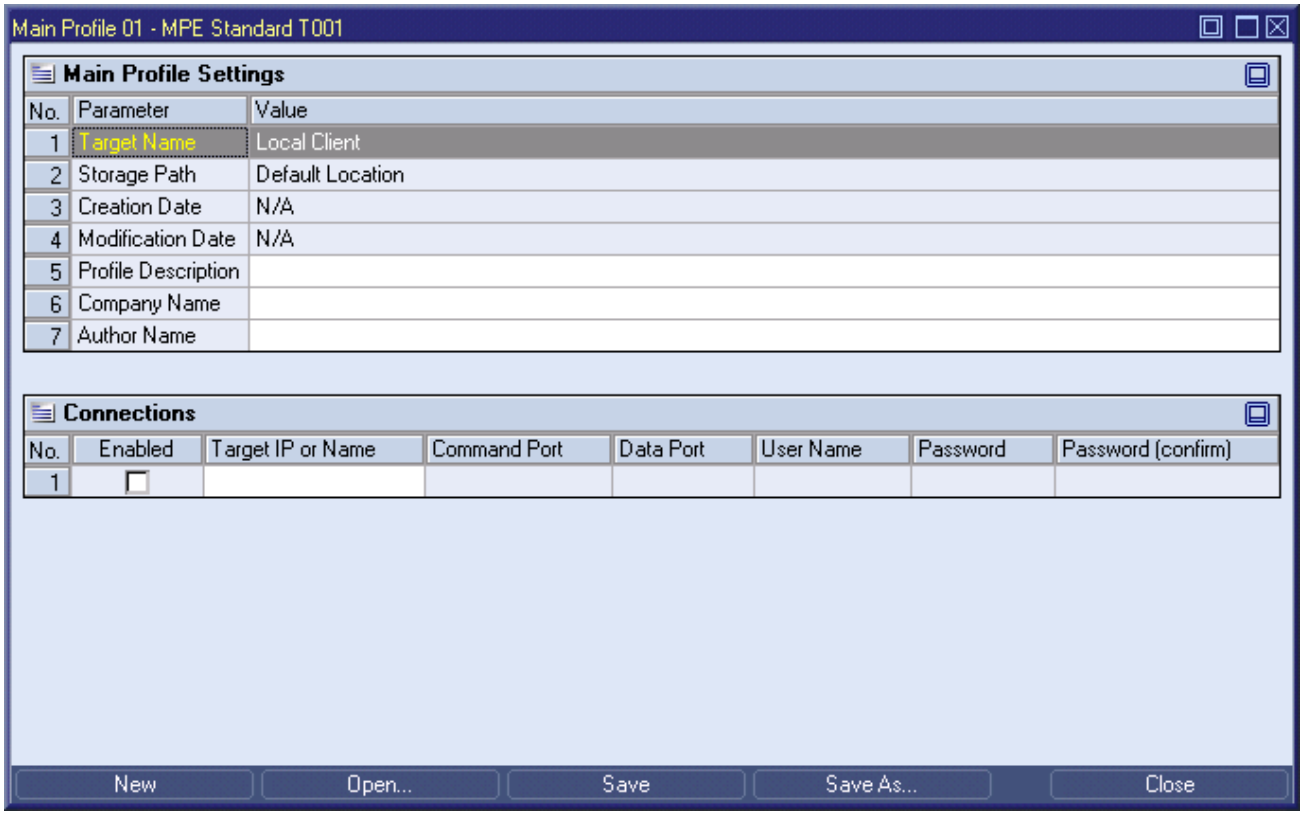


*Figure 19: **MPE Standard T001** with a new Main Profile*

- The Main Profile Settings table at the top of the **MPE Standard T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- Within the Connections table, the actual connections to ***X-Tools Servers*** are configured. In order to configure a new connection, simple insert the IP address to the target ***X-Tools Server*** into the IP Address field. When a connection to the ***X-Tools Server*** at the local computer shall be established, “127.0.0.1” can be entered as IP address.
- By default, the values from the other columns of the Connections table do not need to be modified.
  - In case the ***X-Tools Server*** has been configured to use different command and data ports, these other command and data ports must be entered instead of the default ones.
- In case the anonymous access has been disabled at the ***X-Tools Server***, the proper values for the User Name, Password and Password (confirm) must be provided via the according columns. By default, the ***X-Tools Server*** provides an account with user name “Administrator” and password “administrator”. However, the password for the user “Administrator” also can be changed, in which case the proper other password must be provided.
- The User Name is case-insensitive.

- The Password and Password (confirm) entries are case-sensitive.
- The Save and Save As… buttons are used in order to save the configured Main Profile with all of its settings. All saved Main Profiles are shown by the **MMS Explorer** ("Main -> Files -> *YourLocation* -> Main Profile Files" branch) and can be accessed (e.g. started, stopped, opened, …) from there.
- For a more detailed description of all available functionalities of the **MPE Standard T001**, refer to the reference manual of the "Main Management System", chapter "MPE Standard T001".

Try it out

➔ Create a new Main Profile.

➔ Enter a target IP address of "127.0.0.1" and do not change the default values of the other columns.

➔ Save the Main Profile. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to "Hello Main Profile" and press OK in order to save it to the disk.

The new Main Profile now is displayed below the "Main -> Files -> *Default Location* -> Main Profile Files" branch of the **MMS Explorer**.

Ready

At this point you know how to create a new Main Profile which can be used in order to establish the connections to one or more ***X-Tools Servers***.

6.3.3 Opening of existing Main Profiles

Open the MMS Explorer

- Open the **Main Management System** and ensure that the **MMS Explorer** is displayed at the left side of the **Main Management System** workspace.

Functionality

- Within the **MMS Explorer**, expand the "Main -> Files -> *Default Location* -> Main Profile Files" branch.
- Drag one of the below Main Profiles with the left mouse into the empty space at the right side of the **MMS Explorer**. A **MPE Standard T001** is opened automatically and displays the dropped Main Profile.

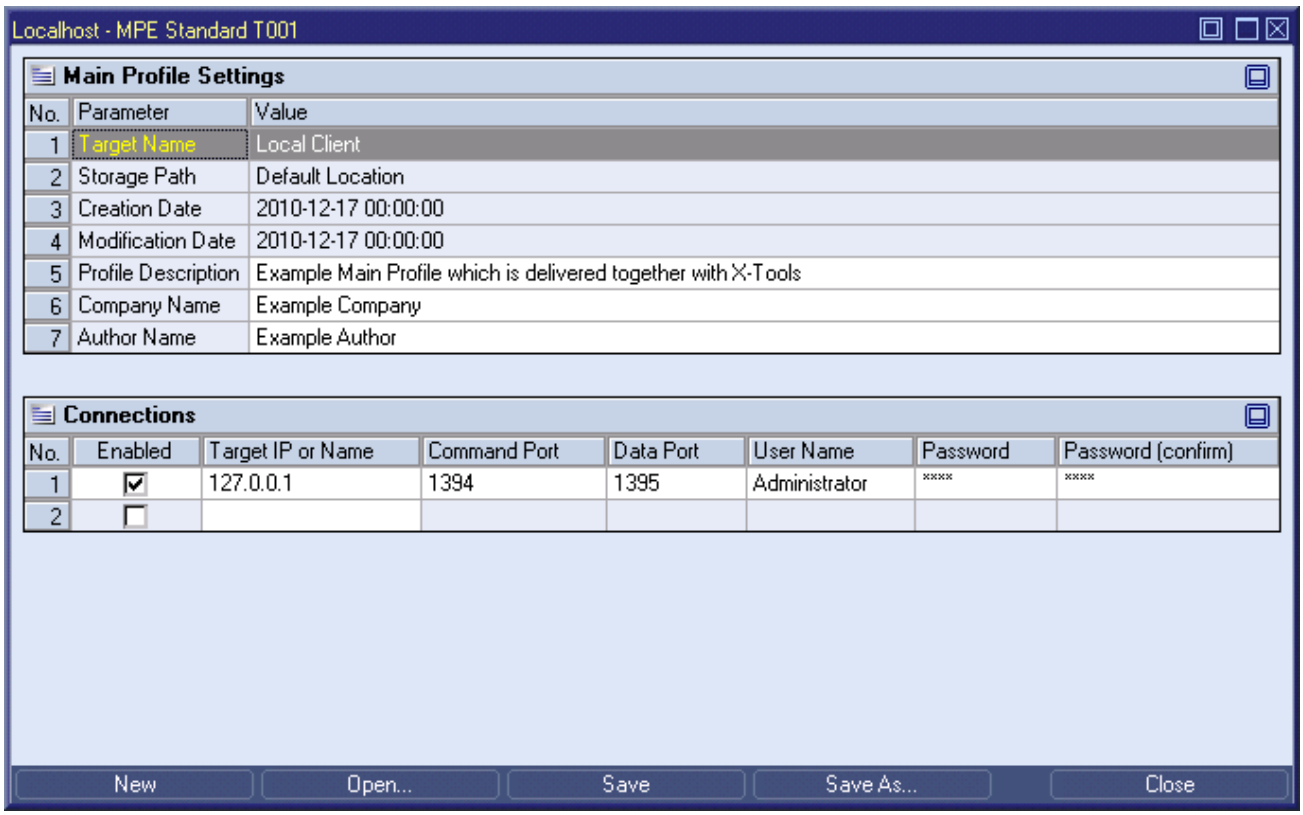

Localhost - MPE Standard T001
Main Profile Settings
No. Parameter Value
1 Target Name Local Client
2 Storage Path Default Location
3 Creation Date 2010-12-17 00:00:00
4 Modification Date 2010-12-17 00:00:00
5 Profile Description Example Main Profile which is delivered together with X-Tools
6 Company Name Example Company
7 Author Name Example Author
Connections
No. Enabled Target IP or Name Command Port Data Port User Name Password Password (confirm)
1 127.0.0.1 1394 1395 Administrator **** ****
2
New Open... Save Save As... Close

*Figure 20: **MPE Standard T001** with an opened Main Profile*

- The opened Main Profile can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited Main Profile with all of its settings.

Try it out

➔ Open an existing Main Profile.

➔ Enter another target IP address and change the Profile Description within the Main Profile Settings table.

➔ Save the Main Profile via the Save button.

The modified Main Profile is still displayed below the “Main -> Files -> *Default Location* -> Main Profile Files” branch of the **MMS Explorer**.

Ready

At this point you know how to open and modify an existing Main Profile.

6.3.4 Starting and Stopping of Main Profiles

Open the MMS Explorer

- Open the **Main Management System** and ensure that the **MMS Explorer** is displayed at the left side of the **Main Management System** workspace.

Functionality

- Within the **MMS Explorer**, expand the “Main -> Files -> *Default Location* -> Main Profile Files” branch.
- Click onto any of the present Main Profiles with the right mouse button in order to open the context menu.
- In order to start the currently selected Main Profile, choose Start from the context menu.
  - Up to one Main Profile can be started at any time. In case a Main Profile is started already and the user starts another one, the currently running Main Profile is stopped automatically and the other Main Profile is being started afterwards.
  - In case there is no Main Profile being started at the moment, the ***X-Tools Client*** does not connect to any ***X-Tools Server***. When there is no ***X-Tools Server*** connected at the moment, all ***X-Tools Server*** related operations (e.g. configuring of measurement, analysis and storage tasks or visualization of data, …) are not available.
- In order to stop the currently running Main Profile, call the context menu of this Main Profile and choose Stop.
  - In case the currently running Main Profile is being stopped, all of the connections to ***X-Tools Servers*** are terminated.

Try it out

➔ Start an existing Main Profile.

The connections to the ***X-Tools Servers*** which are configured within the Main Profile are being established in case the specified ***X-Tools Servers*** are available. The connected ***X-Tools Servers*** appear within all explorer dialogs of the ***X-Tools Client***.

➔ Stop the currently running Main Profile.

All connections to ***X-Tools Servers*** are being terminated and all ***X-Tools Servers*** disappear within all explorer dialogs of the ***X-Tools Client***.

Ready

At this point you know how to start and stop existing Main Profiles in order to establish and terminate the connections to ***X-Tools Servers***.

6.3.5 Securing of the X-Tools Server

The ***X-Tools Server*** provides a user accounts management which can be used in order to maintain multiple users which have different rights at the ***X-Tools Server***. For example, one user may be allowed to edit and start/stop Configuration Files, another one may be allowed to only start/stop existing Configuration Files and a third one may be allowed only to visualize the data which is being provided by the ***X-Tools Server***. Of course there are much more possibilities available – refer to the reference manual of the “Main Management System”, chapter “User Accounts Editor” in order to receive detailed information about the available access rights.

The ***X-Tools Server*** does not protect the current configuration of access rights in any special way and stores all settings into a simple Configuration File (..\Server\User\Basic Settings\User Accounts.uaf.xts). In case the access rights shall be used for security-relevant applications where a user without the according rights must not have any way to assign additional rights to himself, the access to this Configuration File must be limited. This can be done through standard mechanisms of Windows (e.g. by not allowing the Windows user to access the file, which implies that the ***X-Tools Server*** and the ***X-Tools Client*** are running from different Windows user accounts and/or on different computers) and has not been re-implemented by the ***X-Tools Server***.

In case the ***X-Tools Server*** is being configured to be started automatically before a user logs in to Windows, an according service (the ***X-Tools Service***) is being used. This service appears like all other services within the “Services” application of Windows (see “Start” -> “Control Panel” -> “Administrative Tools” -> “Services”). In order to change the Windows user which is being used to start this service (and, therefore, to start the ***X-Tools Server***), open the properties of ***X-Tools Service***, go to the “Log On” tab, select “Log on as this account” and specify the Windows user and password which shall be used to run the service. The chosen user must have write access within the directories of the ***X-Tools Server***.

6.3.6 X-Tools Server Connectivity Dialog

The **X-Tools Server Connectivity** dialog can be called from the context menu of the ***X-Tools Server***, which is available from the task bar of Windows. This dialog can be used in order to check and to edit the communication ports which are being used by the ***X-Tools Server***. In some environments it may happen that the communication ports, which are requested by the ***X-Tools Server***, are being occupied by another application already. In this case, the **X-Tools Server Connectivity** dialog can be used in order to change the communication ports of the ***X-Tools Server***.

After the communication ports of the ***X-Tools Server*** have been changed via the **X-Tools Server Connectivity** dialog, the ***X-Tools Server*** must be re-initialized or re-started.

In case the communication ports of the ***X-Tools Server*** are being changed, also the Main Profiles of all ***X-Tools Clients*** which need to connect to the ***X-Tools Server*** must be updated so that they point to the new communication ports.

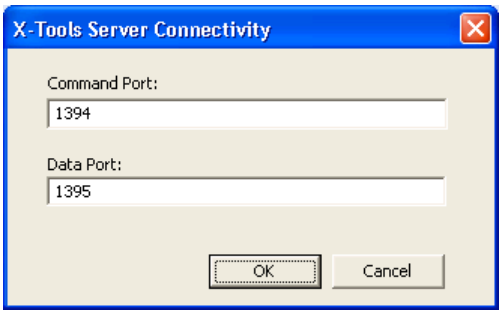


*Figure 21: **X-Tools Server Connectivity** dialog*

6.4 Global Configurations

6.4.1 General Description

In this chapter you will learn how to configure the global options of ***X-Tools***, how to configure the user accounts of ***X-Tools Servers*** and how to visualize the system log files of ***X-Tools***.

This chapter contains the following topics:

- **The Global Options Editor**
- **The User Accounts Editor**
- **The System Log Viewer**

6.4.2 The Global Options Editor

Open the Global Options Editor

- Open the **Main Management System** and ensure that the **MMS Explorer** is displayed at the left side of the **Main Management System** workspace.
- In order to open the **Global Options Editor** for the ***X-Tools Client***, call the context menu for the “Main“ item within the **MMS Explorer** and choose Edit Global Options.
- In order to open the **Global Options Editor** for a connected ***X-Tools Server***, call the context menu for the “Main -> Servers -> *YourServerName*“ item within the **MMS Explorer** and choose Edit Global Options.
- The **Global Options Editor** also can be opened via the Global Options Editor button from the menu bar of the **Main Management System** workspace. In this case, the last time called **Global Options Editor** for the ***X-Tools Client*** or for the connected ***X-Tools Server*** is being opened automatically.
  - The “Main” item from the **MMS Explorer** can be dragged into the empty **Global Options Editor** in order to start editing of the global options of the ***X-Tools Client***.
  - Any of the present “Main -> Servers -> *YourServerName*” items from the **MMS Explorer** can be dragged into the empty **Global Options Editor** in order to start editing of the global options of the ***X-Tools Server***.
- For a more detailed description of all available functionalities of the **Global Options Editor**, refer to the reference manual of the “Main Management System”, chapter “Global Options Editor”.

Functionality

- The Global Option Settings table at the top of the **Global Options Editor** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- Within the Global Options table, the values of all global options can be edited.
- The Save and Save As… buttons are used in order to save the configured global options with all of their settings. Each ***X-Tools Client*** and each ***X-Tools Server*** supports only one Global Options File, thus the user does not need to provide a file name during saving.
  - In case the Global Options File is being stored to the local file system, any possible performed changes of the contained global options are not being used by the ***X-Tools Client*** or the ***X-Tools Server***. The Global Options File must be stored to an ***X-Tools Client*** or to an ***X-Tools Server*** when its configuration shall take into effect.
- In case the ***X-Tools Client*** or any of the currently connected ***X-Tools Servers*** is being dragged into a non-empty **Global Options Editor**, the global options of the dragged item are opened instead of the currently displayed global options.

Try it out

➔ Open the **Global Options Editor** for the ***X-Tools Client***.

➔ Change one or more of the displayed global options.

➔ Save the Global Options File.

In case one or more changes require a restart of the whole application or a restart of single software modules, an according message box is displayed to the user.

➔ Open the **Global Options Editor** for any of the currently connected ***X-Tools Servers***.

➔ Change one or more of the displayed global options.

➔ Save the Global Options File.

In case one or more changes require a restart of the whole application or a restart of single software modules, an according message box is displayed to the user.

Ready

At this point you know how to open and edit the global options of the ***X-Tools Client*** and of the ***X-Tools Server***.

6.4.3 The User Accounts Editor

Open the User Accounts Editor

- Open the **Main Management System** and ensure that the **MMS Explorer** is displayed at the left side of the **Main Management System** workspace.
- In order to open the **User Accounts Editor** for a connected ***X-Tools Server***, call the context menu for the “Main -> Servers -> *YourServerName*“ item within the **MMS Explorer** and choose Edit User Accounts.
- The **User Accounts Editor** also can be opened via the User Accounts Editor button from the menu bar of the **Main Management System** workspace. In this case, an empty **User Accounts Editor** is being opened.
  - Any of the present “Main -> Servers -> *YourServerName*” items from the **MMS Explorer** can be dragged into the empty **User Accounts Editor** in order to start editing of the user accounts of the ***X-Tools Server***.
- For a more detailed description of all available functionalities of the **User Accounts Editor**, refer to the reference manual of the “Main Management System”, chapter “User Accounts Editor”.

Functionality

- The User Accounts Settings table at the top of the **User Accounts Editor** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- Within the User Accounts table, the currently defined user accounts are displayed. In order to configure a new user account, simply insert the user name into the User Name field. In addition to the user name, also a password can be assigned to each user account.
  - In case a user account from the User Accounts table is being selected, the Access Rights, Location Rights and Advanced Rights tables of the **User Accounts Editor** are updated automatically and display the current settings for the currently selected user account.
  - The access rights within the Access Rights, Location Rights and Advanced Rights tables can be configured independently for each defined user account.
- The Save and Save As… buttons are used in order to save the configured user accounts with all of their settings. Each ***X-Tools Server*** supports only one User Accounts File, thus the user does not need to provide a file name during saving.
  - In case the User Accounts File is being stored to the local file system, any possible performed changes of the contained user accounts are not being used by the ***X-Tools Server***. The User Accounts File must be stored to an ***X-Tools Server*** when its configuration shall take into effect.
- In case any of the currently connected ***X-Tools Servers*** is being dragged into a non-empty **User Accounts Editor**, the user accounts of the dragged item are opened instead of the currently displayed user accounts.

Try it out

- ➔ Open the **User Accounts Editor** for any of the currently connected ***X-Tools Servers***.
- ➔ Add a user account with user name “Tutorial” and choose any password.
- ➔ Save the User Accounts File.
- ➔ Change the current Main Profile so that the **X-Tools Client** connects to the chosen **X-Tools Server** with the username “Tutorial” and the assigned password. Stop the Main Profile and restart it so that the made changes within the Main Profile are taken over.

  The ***X-Tools Client*** connects to the ***X-Tools Server*** with the username “Tutorial”. In case the provided combination of user name and password would be wrong, the connection attempt fails and according information is output to the system log of the ***X-Tools Client***.

Ready

At this point you know how to open and edit the user accounts of the ***X-Tools Server***.

6.4.4 The System Log Viewer

Open the System Log Viewer

- Open the **Main Management System** and ensure that the **MMS Explorer** is displayed at the left side of the **Main Management System** workspace.
- In order to open the **System Log Viewer** for the ***X-Tools Client***, call the context menu for the “Main“ item within the **MMS Explorer** and choose View System Log.
- In order to open the **System Log Viewer** for a connected ***X-Tools Server***, call the context menu for the “Main -> Servers -> *YourServer*“ item within the **MMS Explorer** and choose View System Log.
- The **System Log Viewer** also can be opened via the System Log Viewer button from the menu bar of the **Main Management System** workspace. In this case, the last time displayed **System Log Viewer** for the ***X-Tools Client*** or for the connected ***X-Tools Server*** is being opened automatically.
  - The “Main” item from the **MMS Explorer** can be dragged into the empty **System Log Viewer** in order to start viewing of the system log of the ***X-Tools Client***.
  - Any of the present “Main -> Servers -> *YourServer*” items from the **MMS Explorer** can be dragged into the empty **System Log Viewer** in order to start viewing of the system log of the ***X-Tools Server***.
- For a more detailed description of all available functionalities of the **System Log Viewer**, refer to the reference manual of the “Main Management System”, chapter “System Log Viewer”.

Functionality

- The System Log Settings table at the top of the **System Log Viewer** displays some general information about the currently displayed log entries and allows to define filters in order to configure which system log entries shall be displayed and which ones not.
- Within the System Log Entries table, all of the log entries from the currently chosen sources and which match the currently configured filters are being displayed.
- The Apply button is used in order to take over the changes which have been made within the System Log Settings table.
- In case the ***X-Tools Client*** or any of the currently connected ***X-Tools Servers*** is being dragged into a non-empty **System Log Viewer**, the system log entries of the dragged item are added to the currently displayed system log entries.
  - <Shift> can be used when releasing the mouse button in order to replace the currently displayed system log entries with the system log entries of the dragged item and to determine the currently dragged item as the one and only source for to-be-displayed system log entries.

Try it out

➔ Open the **System Log Viewer** for the ***X-Tools Client***.

The log entries of the ***X-Tools Client*** are displayed.

➔ Change one or more of the available visualization filters.

Only the log entries which apply to the chosen filters are displayed.

➔ Drag one of the currently connected ***X-Tools Servers*** so that its log entries are being displayed in addition to the already displayed log entries of the ***X-Tools Client***.

The log entries of the ***X-Tools Client*** and of the chosen ***X-Tools Server*** are being displayed together.

➔ Drag the ***X-Tools Client*** so that its log entries replace all of the currently displayed log entries.

Only the log entries of the ***X-Tools Client*** are displayed again.

Ready

At this point you know how to open system log of the ***X-Tools Client*** and of the ***X-Tools Server***.

6.5 Configuration File Locations

6.5.1 General Description

In this chapter you will learn how Configuration File locations can be used in order to switch the complete configuration of an ***X-Tools Server*** with only a few clicks from one configuration to another configuration.

Each Configuration File location contains a complete set of Interface Profiles, Device Profiles, Device Setups, Monitoring Views, Analyzing Models, Analyzing Scripts, Storage Profiles and Loading Profiles. When a Configuration File location is added to the ***X-Tools Server***, all of the contained configurations take into effect - according to the latest configuration, any of the contained Configuration Files can be started.

Configuration File locations can be used for various applications, e.g. in case of mobile systems which have to be configured differently at each measurement point, in case specialized configurations are wanted for error tracking or in case the authors of the tutorial want to provide a large range of example configurations which can show the capabilities of ***X-Tools*** in addition to the descriptions within the tutorial.

This chapter contains the following topics:

- **Adding of a Configuration File Location**
- **Removing of a Configuration File Location**

6.5.2 Adding of a Configuration File Location

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Functionality

- Expand the “Main -> Servers -> *YourServer*” branch of the **DMS Explorer**.
- In order to add another Configuration File location, call the context menu for the “Files“ item within the **DMS Explorer** and choose Add Location….
- Within the upcoming **Add Location** dialog, choose any symbolic name for your new Configuration File location and specify the path to this location at the disk.
  - In case the path to the location at the disk is chosen first, a default symbolic name is provided automatically.
- Press OK in order to add the specified location to the ***X-Tools Server***.
- After another Configuration File location has been added successfully, the ***X-Tools Server*** automatically creates all of the necessary sub-directories. In case the added Configuration File location contains some Configuration Files already, the Configuration Files which are marked as to-be-started are being started automatically.
- The Add Location… context menu item is available not only within the **DMS Explorer** but also within the **MTS Explorer**, **ANS Explorer** and **STS Explorer**. As Configuration File locations are valid system-wide, they can be added via any of the above explorer trees and they become available within all explorer trees in case they are added via one of them.

Try it out

➔ Add the “..\CMS X-Tools Pro\Server\User\Example Location” directory as additional Configuration File location.

All of the Configuration Files from the example location become visible within the explorer trees of the ***X-Tools Server***.

➔ Start the Interface Profile Simulation T001 from the example location.

The according simulated devices are being added to the **DMS Explorer** and are visible under the Main -> Servers -> *YourServer* -> Interfaces -> Simulation T001.

➔ Start a Device Profile from the example location for one of the devices which are being simulated.

The according data are being added to the **MDS Explorer**.

Ready

At this point you know how to add Configuration File locations to the ***X-Tools Server***.

6.5.3 Removing of a Configuration File Location

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Functionality

- Expand the “Main -> Servers -> *YourServer* -> Files” branch of the **DMS Explorer**.
- In order to remove an existing Configuration File location, call the context menu for any of the present Configuration File locations within the **DMS Explorer** and choose Remove Location.
- After confirming of the upcoming dialog, the chosen Configuration File location is being removed from all explorer trees of the ***X-Tools Server***. The Configuration Files below the removed Configuration File location are being stopped by this operation.
- In addition to the Remove Location context menu item, also Delete Location is available. Where Remove Location only removes the specified Configuration File location from the knowledge of the ***X-Tools Server***, Delete Location actually deletes all of the files of the location from the disk.

Try it out

➔ Remove any of the present Configuration File locations.

All of the Configuration Files from the specified location are being removed from the explorer trees of the ***X-Tools Server***. All of the Configuration Files from the removed Configuration File location are being stopped.

Ready

At this point you know how to remove and delete Configuration File locations from the ***X-Tools Server*** and from the disk.

6.6 Simulated Devices and Interface Profiles

6.6.1 General Description

In this chapter you will learn how to attach simulated devices to the ***X-Tools Server*** via Interface Profiles for the Simulation interface. Simulated devices are used in order to show how to use the system. For ***X-Tools***, there is no difference in the configuration or usage of data between simulated and real devices. Thus, all operations which are performed on simulated devices (e.g. creation of Device Profiles, recording of measurement data, performing of analysis, storage of data, …) are done in exactly the same way with real devices, and all functionalities of ***X-Tools*** can be tried out without any real hardware attached.

Examples of Interface Profiles are found within the example Configuration File location which is being delivered together with ***X-Tools***.

This chapter contains the following topics:

- **Creation of a new Interface Profile for the Simulation Interface**
- **Opening of existing Interface Profiles**
- **Starting and Stopping of Interface Profiles**

6.6.2 Creation of a new Interface Profile for the Simulation Interface

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Functionality

- Method 1 for the creation of a new Interface Profile:
  - Within the **DMS Explorer**, expand the “Main -> Editors -> Interface Profile Editors” branch.
  - Drag the **IPE Simulation T001** with the left mouse into the empty space at the right side of the **DMS Explorer**. An empty **IPE Simulation T001** is opened automatically.
  - Press the New button in order to create a new Interface Profile. The new Interface Profile is displayed by the **IPE Simulation T001** with default values.
  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment. It can be used for the creation and configuration of Interface Profiles for cases where the target ***X-Tools Server*** is not present at the moment.
- Method 2 for the creation of a new Interface Profile:
  - Within the **DMS Explorer**, expand the “Main -> Servers -> *YourServer* -> Interfaces” branch.
  - Click onto the present “Simulation T001” item with the right mouse button in order to open the context menu.
  - In order to create a new Interface Profile for the currently selected interface, choose New Interface Profile from the context menu.
  - This method works only in case the target ***X-Tools Server*** is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new Interface Profile.
- The following screenshot shows an example of a new Interface Profile.

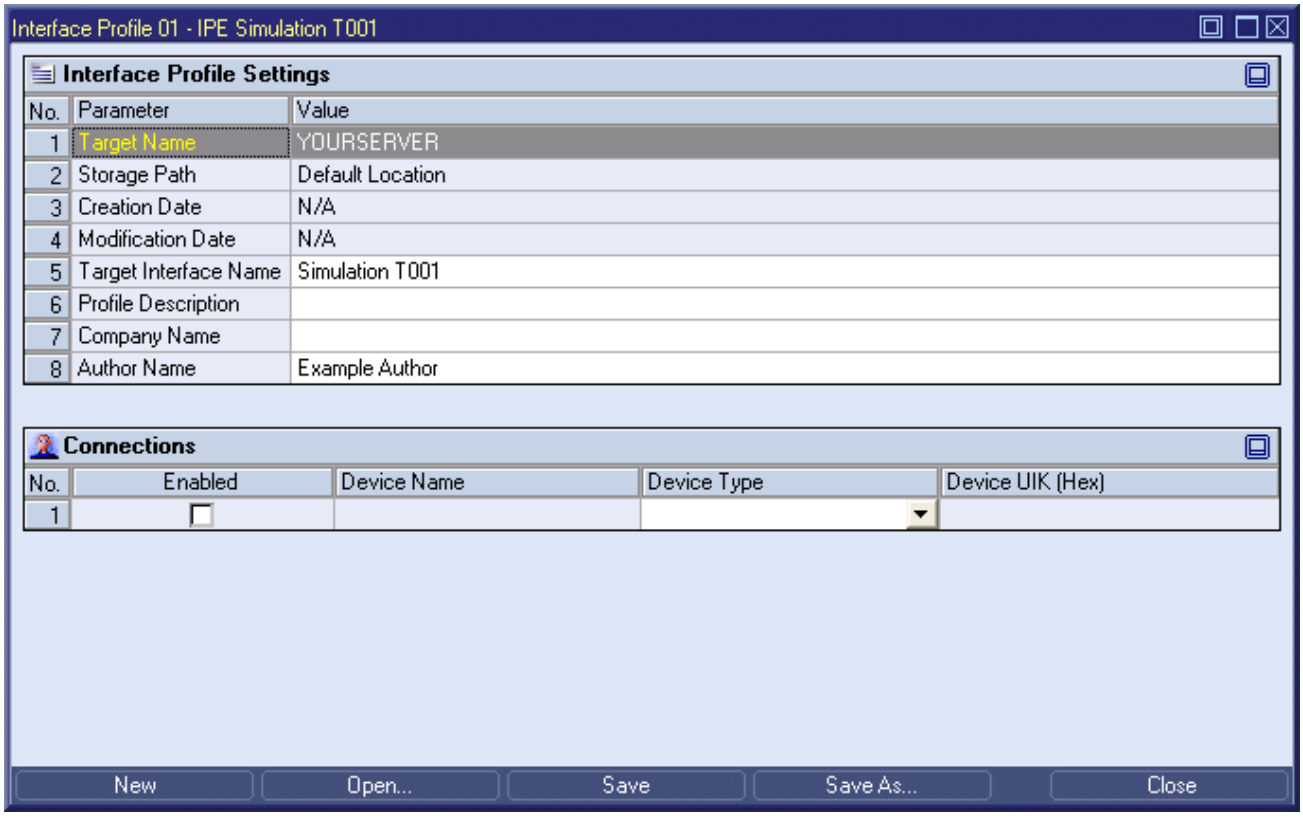


*Figure 22: **IPE Simulation T001** with a new Interface Profile*

- The Interface Profile Settings table at the top of the **IPE Simulation T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.

- Within the Connections table, the to-be-simulated devices are configured. In order to configure a new device, simple choose the desired device type from the Device Type field.
- By default, the values from the other columns of the Connections table do not need to be modified.
  - The Device Name and Device UIK fields can be used in order to change the default name and UIK of each device which will be simulated.
- The Save and Save As… buttons are used in order to save the configured Interface Profile with all of its settings. All saved Interface Profiles are shown by the **DMS Explorer** ("Main -> Servers -> *YourServer* -> *YourLocation* -> Interface Profile Files" branch) and can be accessed (e.g. started, stopped, opened, …) from there.
- For a more detailed description of all available functionalities of the **IPE Simulation T001** and the other **Interface Profile Editors**, refer to the reference manual of the "Device Management System", chapter "Interface Profile Editors".

Try it out

➔ Create a new Interface Profile.

➔ Configure the simulation of one ION AnalogInput T001, two ION VIB-A T001 and one ION SIMATIC S7-300 T001.

➔ Save the Interface Profile. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to "Hello Interface Profile" and press OK in order to save it to the disk.

The new Interface Profile now is displayed below the "Main -> Servers -> *YourServer* -> *YourLocation* -> Interface Profile Files" branch of the **DMS Explorer**.

➔ Take a look onto the Interface Profiles which are provided via the example location (see point 6.5) in order to see various application examples about the capabilities and usage of Interface Profiles.

Ready

At this point you know how to create a new Interface Profile which can be used in order to simulate any of the devices which are supported by the ***X-Tools Server***.

6.6.3 Opening of existing Interface Profiles

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Functionality

- Within the **DMS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Interface Profile Files” branch.
- Drag one of the below Interface Profiles with the left mouse into the empty space at the right side of the **DMS Explorer**. An **Interface Profile Editor** of proper type is opened automatically and displays the dropped Interface Profile.

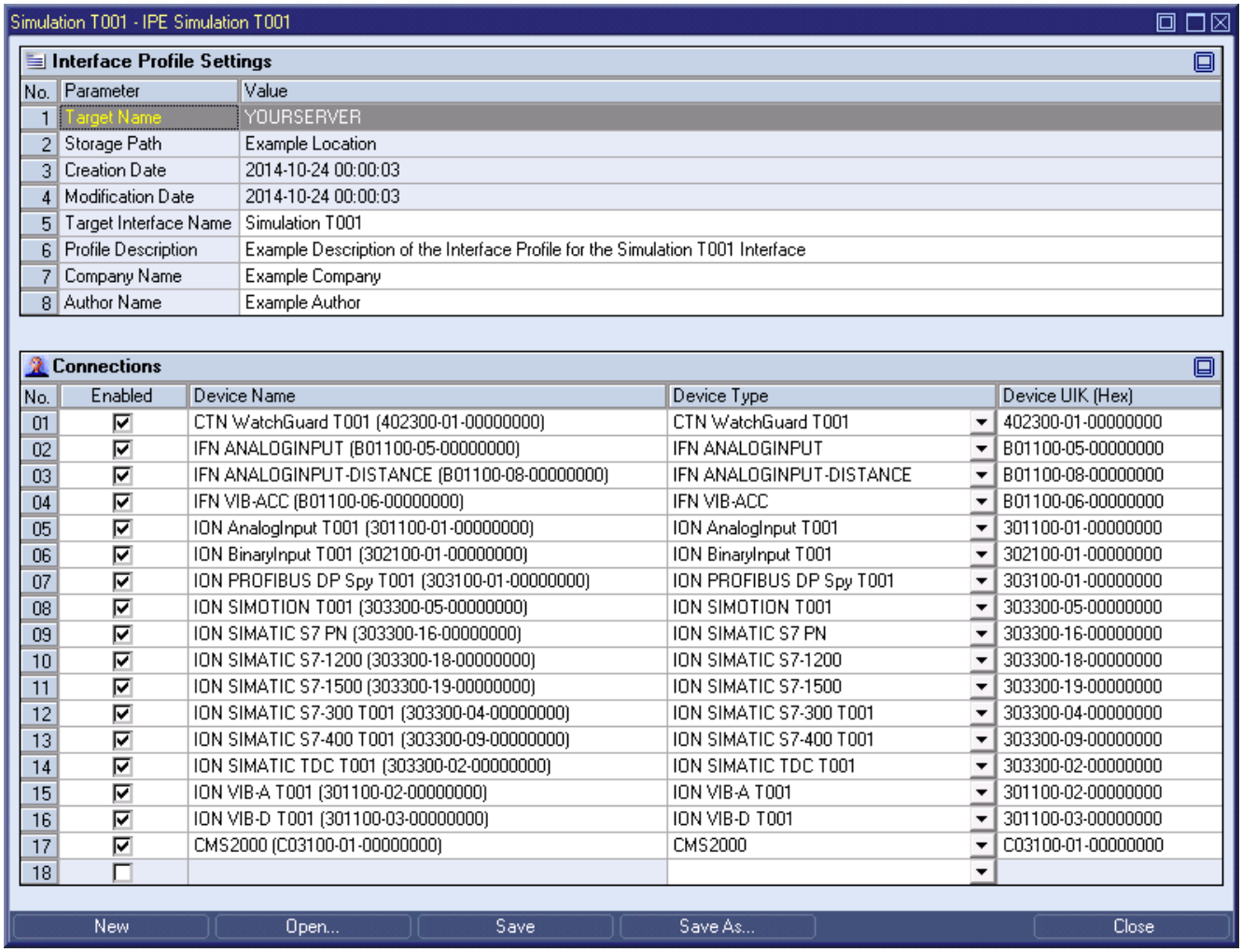

Simulation T001 - IPE Simulation T001
Interface Profile Settings
No. Parameter Value
1 Target Name YOURSERVER
2 Storage Path Example Location
3 Creation Date 2014-10-24 00:00:03
4 Modification Date 2014-10-24 00:00:03
5 Target Interface Name Simulation T001
6 Profile Description Example Description of the Interface Profile for the Simulation T001 Interface
7 Company Name Example Company
8 Author Name Example Author
Connections
No. Enabled Device Name Device Type Device UIK (Hex)
01 CTN WatchGuard T001 (402300-01-00000000) CTN WatchGuard T001 402300-01-00000000
02 IFN ANALOGINPUT (B01100-05-00000000) IFN ANALOGINPUT B01100-05-00000000
03 IFN ANALOGINPUT-DISTANCE (B01100-08-00000000) IFN ANALOGINPUT-DISTANCE B01100-08-00000000
04 IFN VIB-ACC (B01100-06-00000000) IFN VIB-ACC B01100-06-00000000
05 ION AnalogInput T001 (301100-01-00000000) ION AnalogInput T001 301100-01-00000000
06 ION BinaryInput T001 (302100-01-00000000) ION BinaryInput T001 302100-01-00000000
07 ION PROFIBUS DP Spy T001 (303100-01-00000000) ION PROFIBUS DP Spy T001 303100-01-00000000
08 ION SIMOTION T001 (303300-05-00000000) ION SIMOTION T001 303300-05-00000000
09 ION SIMATIC S7 PN (303300-16-00000000) ION SIMATIC S7 PN 303300-16-00000000
10 ION SIMATIC S7-1200 (303300-18-00000000) ION SIMATIC S7-1200 303300-18-00000000
11 ION SIMATIC S7-1500 (303300-19-00000000) ION SIMATIC S7-1500 303300-19-00000000
12 ION SIMATIC S7-300 T001 (303300-04-00000000) ION SIMATIC S7-300 T001 303300-04-00000000
13 ION SIMATIC S7-400 T001 (303300-09-00000000) ION SIMATIC S7-400 T001 303300-09-00000000
14 ION SIMATIC TDC T001 (303300-02-00000000) ION SIMATIC TDC T001 303300-02-00000000
15 ION VIB-A T001 (301100-02-00000000) ION VIB-A T001 301100-02-00000000
16 ION VIB-D T001 (301100-03-00000000) ION VIB-D T001 301100-03-00000000
17 CMS2000 (C03100-01-00000000) CMS2000 C03100-01-00000000
18
New
Open...
Save
Save As...
Close

*Figure 23: **IPE Simulation T001** with an opened Interface Profile*

- The opened Interface Profile can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited Interface Profile with all of its settings.

Try it out

➔ Open an existing Interface Profile.

➔ Enter another device type and change the Profile Description within the Interface Profile Settings table.

➔ Save the Interface Profile via the Save button.

The modified Interface Profile is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Interface Profile Files” branch of the **DMS Explorer**.

Ready

At this point you know how to open and modify an existing Interface Profile.

6.6.4 Starting and Stopping of Interface Profiles

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Functionality

- Within the **DMS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Interface Profile Files” branch.
- Click onto any of the present Interface Profiles with the right mouse button in order to open the context menu.
- In order to start the currently selected Interface Profile, choose Start from the context menu.
  - Up to one Interface Profile can be started at any time for each of the present interfaces of an ***X-Tools Server***. In case an Interface Profile is started already for an interface and the user starts another one for the same interface, the currently running Interface Profile is stopped automatically and the other Interface Profile is being started afterwards.
  - In case there is no Interface Profile being started at the moment for a certain interface, no devices from this interface are available to the ***X-Tools Server***. The only exception to this rule is the IEEE1394 interface, which detects the currently attached devices via Plug&Play and does not need any according Interface Profile.
- In order to stop a currently running Interface Profile, call the context menu of this Interface Profile and choose Stop.
  - In case the currently running Interface Profile is being stopped, all of its currently present devices are disconnected and not available to the ***X-Tools Server*** any more.
- There are several conditions in which an Interface Profile can not run after it has been started, e.g. in case the target interface is not available at the moment. When an Interface Profile can not run after it has been started, it is being suspended and automatically resumed as soon as the error condition has disappeared.
- In case at least one Interface Profile has been started by the user, the status bar of the ***X-Tools Client*** displays the current status of Interface Profiles via the according colored icon. The tool tip of this icon provides detailed information about the number of currently started and suspended Interface Profiles.
  - Note that the status bar always displays the status of exactly one ***X-Tools Server***, thus also the icon for the current status of Interface Profiles displays the status of only the currently chosen ***X-Tools Server***.

Try it out

➔ Start an existing Interface Profile for the Simulation interface.

The devices which are configured within the Interface Profile are being simulated and appear below the “Main -> Servers -> *YourServer* -> Interfaces -> Simulation T001” branch immediately.

➔ Stop the currently running Interface Profile for the Simulation interface.

The simulation of devices is being stopped and all simulated devices disappear within the **DMS Explorer** of the ***X-Tools Server***.

Ready

At this point you know how to start and stop existing Interface Profiles for the Simulation interface in order to start and stop the simulation of devices.

6.7 Device Configuration and Device Profiles

6.7.1 General Description

In this chapter you will learn how to configure devices via Device Profiles. A Device Profile contains all of the information which is needed in order to perform the measurement with a device. This information includes instructions for the target device as well as data names, normalization factors, buffer sizes, …. Multiple Device Profiles can be created for each device, but only one Device Profile per device can be started at a time. The started Device Profile of each device is the one whose configuration is sent to the device in order to configure it. In addition, the data which are defined within the started Device Profile are available to all other modules of ***X-Tools*** via the **MDS Explorer**.

For each device type a specialized **Device Profile Editor** is provided by the ***X-Tools Client***. The specialized **Device Profile Editor** of each device type provides all of the functionality which is needed in order to edit Device Profiles as convenient as possible.

The following explanations refer to simulated devices. However, the handling of Device Profiles is identical between simulated and real devices and therefore the following operations can be applied to real devices as well.

Examples of Device Profiles are found within the example Configuration File location which is being delivered together with ***X-Tools***.

This chapter contains the following topics:

- **Creation of a new Device Profile for an ION AnalogInput T001**
- **Opening of existing Device Profiles**
- **Starting and Stopping of Device Profiles**

6.7.2 Creation of a new Device Profile for an ION AnalogInput T001

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.
- Ensure that you have at least one ION AnalogInput T001 simulated below the “Main -> Servers -> *Your-Server* -> Interfaces -> Simulation T001” branch of the **DMS Explorer**.

Functionality

- Method 1 for the creation of a new Device Profile:
  - Within the **DMS Explorer**, expand the “Main -> Editors -> Device Profile Editors” branch.
  - Drag the **DPE ION AnalogInput T001** with the left mouse into the empty space at the right side of the **DMS Explorer**. An empty **DPE ION AnalogInput T001** is opened automatically.
  - Press the New button in order to create a new Device Profile. The new Device Profile is displayed by the **DPE ION AnalogInput T001** with default values.
  - The new Device Profile must be told for which target device is shall be used. In order to specify a target device, either enter the name of the target device into the Target Device Name cell of the Device Profile Settings table or Drag&Drop the desired target device from the **DMS Explorer** onto the Target Device Name cell.
  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment and/or in case the desired target device is not available at the moment. It can be used for the creation and configuration of Device Profiles for cases where the target ***X-Tools Server*** and/or the target device are not present at the moment.

- Method 2 for the creation of a new Device Profile:
  - Within the **DMS Explorer**, expand the “Main -> Servers -> *YourServer* -> Interfaces -> Simulation T001” branch.
  - Click onto a present ION AnalogInput T001 with the right mouse button in order to open the context menu.
  - In order to create a new Device Profile for the currently selected device, choose New Device Profile from the context menu.
  - This method works only in case the target ***X-Tools Server*** and the target device are present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** and the target device of the new Device Profile.
- The following screenshot shows an example of a new Device Profile.

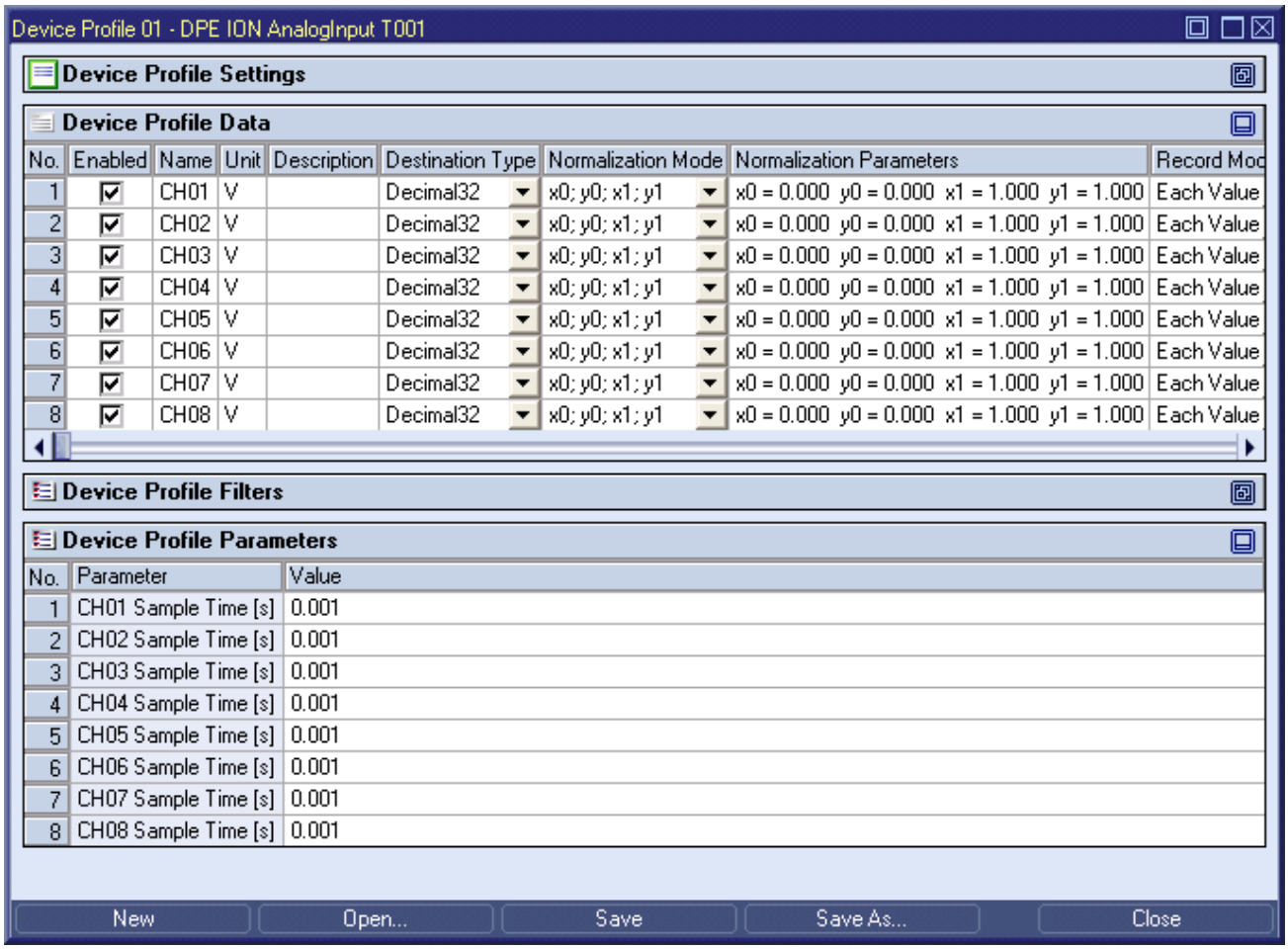

Device Profile 01 - DPE ION AnalogInput T001
Device Profile Settings
Device Profile Data
No. Enabled Name Unit Description Destination Type Normalization Mode Normalization Parameters Record Mod
1 CH01 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
2 CH02 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
3 CH03 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
4 CH04 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
5 CH05 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
6 CH06 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
7 CH07 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
8 CH08 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
Device Profile Filters
Device Profile Parameters
No. Parameter Value
1 CH01 Sample Time [s] 0.001
2 CH02 Sample Time [s] 0.001
3 CH03 Sample Time [s] 0.001
4 CH04 Sample Time [s] 0.001
5 CH05 Sample Time [s] 0.001
6 CH06 Sample Time [s] 0.001
7 CH07 Sample Time [s] 0.001
8 CH08 Sample Time [s] 0.001
New
Open...
Save
Save As...
Close

*Figure 24: **DPE ION AnalogInput T001** with a new Device Profile*

- The Device Profile Settings table at the top of the **DPE ION AnalogInput T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company. In addition, Device Profile related settings like the target device name, the data name prefix or the data supervision time are configured via the Device Profile Settings table.
- Within the Device Profile Data table, the 8 input channels of the ION AnalogInput T001 can be configured. There, all of the information about each channel is being defined, like the data name, the data description, the normalization, the buffer size or the record mode.
- Within the Device Profile Filters table, the filter coefficients of all 8 input channels can be configured.
- Within the Device Profile Parameters table, the sample rates of all 8 input channels can be configured.
- The default values and settings which are provided for a new Device Profile can be used for real measurement already and without any modifications. Depending to each measurement task, a modification of the default values and settings may be necessary or not.

- The Save and Save As… buttons are used in order to save the configured Device Profile with all of its settings. All saved Device Profiles are shown by the **DMS Explorer** (“Main -> Servers -> *YourServer* -> *YourLocation* -> Device Profile Files” branch) and can be accessed (e.g. started, stopped, opened, …) from there.
- For a more detailed description of all available functionalities of the **DPE ION AnalogInput T001** and the other **Device Profile Editors**, refer to the reference manual of the “Device Management System”, chapter “Device Profile Editors”.

Try it out

➔ Create a new Device Profile.

➔ Change some names and descriptions within the Device Profile Data table.

➔ Save the Device Profile. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to “Hello Device Profile” and press OK in order to save it to the disk.

The new Device Profile now is displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Device Profile Files” branch of the **DMS Explorer**.

➔ Take a look onto the Device Profiles which are provided via the example location (see point 6.5) in order to see various application examples about the capabilities and usage of Device Profiles.

Ready

At this point you know how to create a new Device Profile which can be used in order to configure an ION AnalogInput T001.

6.7.3 Opening of existing Device Profiles

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Functionality

- Within the **DMS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Device Profile Files” branch.
- Drag one of the below Device Profiles with the left mouse into the empty space at the right side of the **DMS Explorer**. A **Device Profile Editor** of proper type is opened automatically and displays the dropped Device Profile.

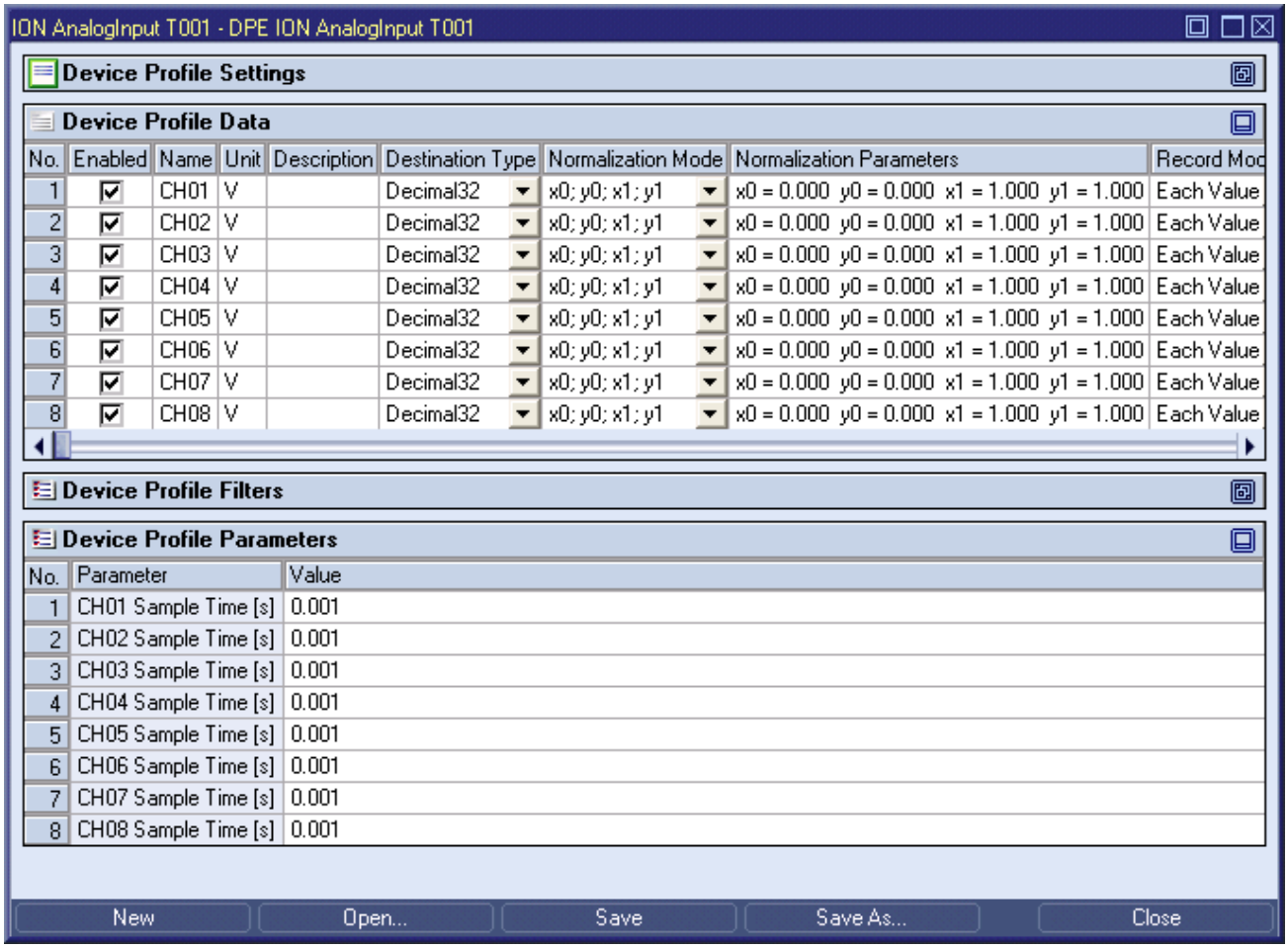

ION AnalogInput T001 - DPE ION AnalogInput T001
Device Profile Settings
Device Profile Data
No. Enabled Name Unit Description Destination Type Normalization Mode Normalization Parameters Record Mod
1 CH01 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
2 CH02 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
3 CH03 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
4 CH04 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
5 CH05 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
6 CH06 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
7 CH07 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
8 CH08 V Decimal32 x0; y0; x1; y1 x0 = 0.000 y0 = 0.000 x1 = 1.000 y1 = 1.000 Each Value
Device Profile Filters
Device Profile Parameters
No. Parameter Value
1 CH01 Sample Time [s] 0.001
2 CH02 Sample Time [s] 0.001
3 CH03 Sample Time [s] 0.001
4 CH04 Sample Time [s] 0.001
5 CH05 Sample Time [s] 0.001
6 CH06 Sample Time [s] 0.001
7 CH07 Sample Time [s] 0.001
8 CH08 Sample Time [s] 0.001
New
Open...
Save
Save As...
Close

*Figure 25: **DPE ION AnalogInput T001** with an opened Device Profile*

- The opened Device Profile can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited Device Profile with all of its settings.

Try it out

➔ Open an existing Device Profile.

➔ Enter another Target Device Name and change the Profile Description within the Device Profile Settings table.

➔ Save the Device Profile via the Save button.

The modified Device Profile is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Device Profile Files” branch of the **DMS Explorer**.

Ready

At this point you know how to open and modify an existing Device Profile.

6.7.4 Starting and Stopping of Device Profiles

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.
- Ensure that you have at least one ION AnalogInput T001 simulated below the “Main -> Servers -> *YourServer* -> Interfaces -> Simulation T001” branch of the **DMS Explorer**.

Functionality

- Within the **DMS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Device Profile Files” branch.
- Click onto any of the present Device Profiles with the right mouse button in order to open the context menu.
- In order to start the currently selected Device Profile, choose Start from the context menu.
  - Up to one Device Profile can be started at any time for each of the present devices of an ***X-Tools Server***. In case a Device Profile is started already for a device and the user starts another one for the same device, the currently running Device Profile is stopped automatically and the other Device Profile is being started afterwards.
  - In case there is no Device Profile being started at the moment for a certain device, the device is not configured and does not deliver any measurement data to the ***X-Tools Server***.
- The configured data of each started Device Profile is found within the **MDS Explorer** below its “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch. From there, the data can be moved to all systems of ***X-Tools*** for further processing via Drag&Drop.
- In order to stop a currently running Device Profile, call the context menu of this Device Profile and choose Stop.
  - In case the currently running Device Profile is being stopped, the device stops to deliver further measurement data to the ***X-Tools Server***.
- There are several conditions in which a Device Profile can not run after it has been started, e.g. in case the target device is not available at the moment or in case the configured data names are already in use. When a Device Profile can not run after it has been started, it is being suspended and automatically resumed as soon as the error condition has disappeared.
- In case at least one Device Profile has been started by the user, the status bar of the ***X-Tools Client*** displays the current status of Device Profiles via the according colored icon. The tool tip of this icon provides detailed information about the number of currently started and suspended Device Profiles.
  - Note that the status bar always displays the status of exactly one ***X-Tools Server***, thus also the icon for the current status of Device Profiles displays the status of only the currently chosen ***X-Tools Server***.

Try it out

➔ Start the Interface Profile Simulation T001 for the Simulation interface.

➔ Start an existing Device Profile for the simulated ION AnalogInput T001.

The data which has been configured within the started Device Profile becomes available within the **MDS Explorer**.

➔ Stop the currently running Device Profile for the simulated ION AnalogInput T001.

The data which has been configured within the started Device Profile is being removed from the **MDS Explorer**.

Ready

At this point you know how to start and stop existing Device Profiles in order to start and stop the measurement of data via a device.

6.8 Visualization of Online Data and Monitoring Views

6.8.1 General Description

In this chapter you will learn how to visualize online data from the **MDS Explorer** with the **Monitoring System** using the most basic functionalities and how to create and maintain Monitoring Views. A Monitoring View contains the exact configuration of the visualized curves, including all settings. It is used in order to save all relevant settings and to restore them with a single Drag&Drop operation.

Examples of Monitoring Views are found within the example Configuration File location which is being delivered together with ***X-Tools***.

This chapter contains the following topics:

- **Creation of a new Monitoring View**
- **Opening of existing Monitoring Views**
- **Starting and Stopping of Monitoring Views**

6.8.2 Creation of a new Monitoring View

Open the MTS Explorer

- Open the **Monitoring System** and ensure that the **MTS Explorer** is displayed at the left side of the **Monitoring System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch of the **MDS Explorer**.

Functionality

- Method 1 for the creation of a new Monitoring View:
  - Within the **MTS Explorer**, expand the “Main -> Editors -> Monitoring View Editors” branch.
  - Drag the **MVE Standard T001** with the left mouse into the empty space at the right side of the **MTS Explorer**. An empty **MVE Standard T001** is opened automatically.
  - Press the New button in order to create a new Monitoring View. The new Monitoring View is displayed by the **MVE Standard T001** with default values. It does not contain any charts yet.
  - Within the **MTS Explorer**, expand the “Main -> Editors -> Monitoring Charts” branch.
  - Drag the **MTC yt T001** with the left mouse into the opened Monitoring View. An empty **MTC yt T001** is opened automatically.
    - In case the **MTC yt T001** is dragged from the **MTS Explorer** into the empty **Monitoring System**, the **MVE Standard T001** is opened automatically and also opens an empty **MTC yt T001**.
  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment and/or in case the desired target data is not available at the moment. It can be used for the creation and pre-configuration of Monitoring Views for cases where the target ***X-Tools Server*** and/or the target data are not present at the moment.
- Method 2 for the creation of a new Monitoring View:
  - Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch.
  - Drag&Drop the desired target data into the empty **Monitoring System**. The **MVE Standard T001** is opened automatically and also opens the proper **Monitoring Chart**. In addition, the dropped data is added to and visualized by the opened **Monitoring Chart** automatically.
  - This method works only in case the target ***X-Tools Server*** and the target data are present but it is more convenient and faster as it automatically opens all windows.

- The following screenshot shows an example of a new Monitoring View.

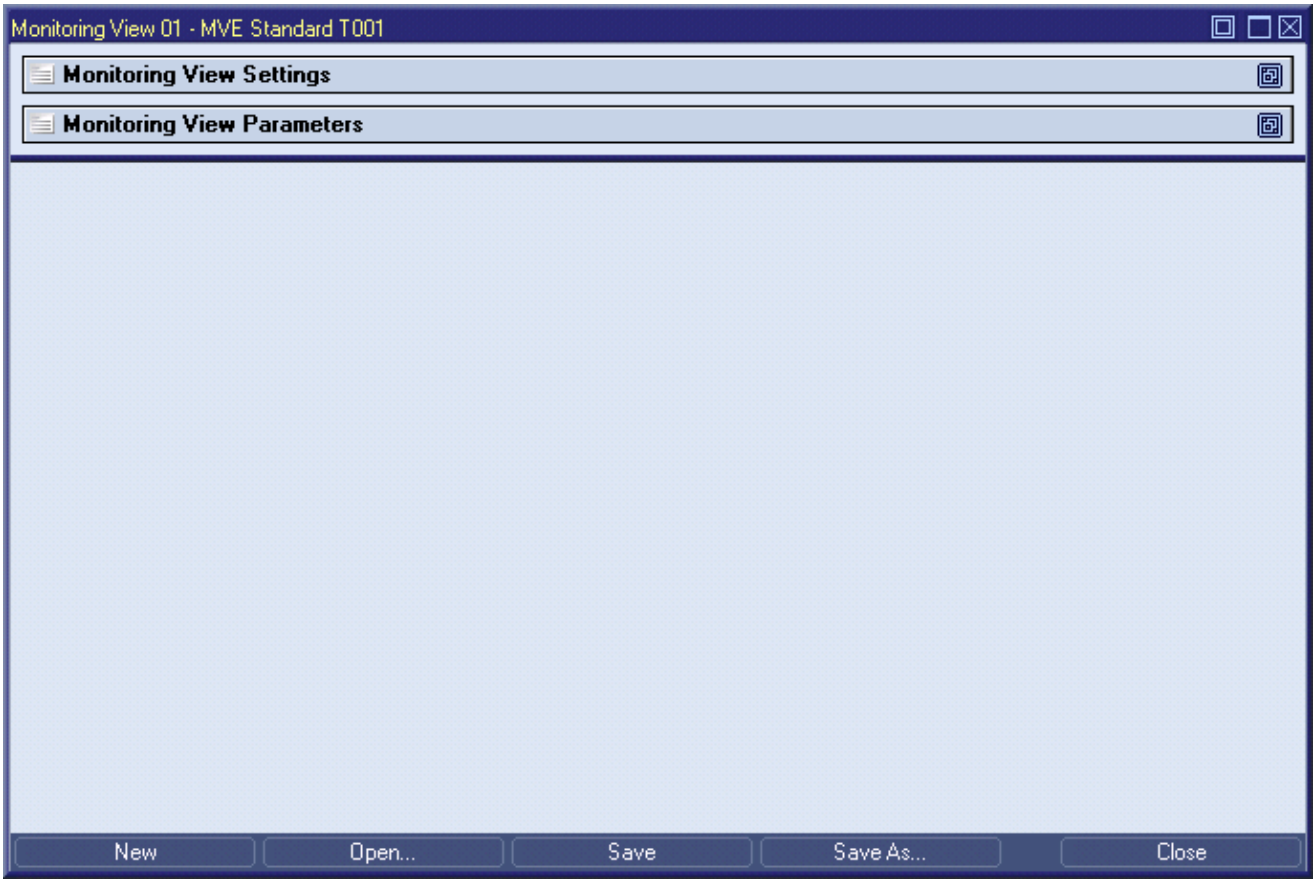


*Figure 26: **MVE Standard T001** with a new Monitoring View*

- The Monitoring View Settings table at the top of the **MVE Standard T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- Within the Monitoring View Parameters table, additional parameters like the Time Domain can be configured.
- In order to visualize any of the currently available online data, drag the desired data from the “Main -> Servers -> *YourServer* -> Online Data” branch of the **MDS Explorer** into an empty **MVE Standard T001** or into an opened **Monitoring Chart**. Up to 32 data can be put into one **Monitoring Chart**.
- Several mechanisms are available for the changing of the visualization of the displayed data:
  - When the left mouse button is being pressed, zooming can be performed.
    - Press the left mouse button at an axis (e.g. y-axis, t-axis or x-axis) and move the mouse cursor in order to shrink or stretch the scaling.
    - Press the left mouse button within the curve area and move the mouse cursor in order to define the area into which it shall be zoomed. <Shift> or <x> can be pressed in order to zoom only into x-direction, <Ctrl> or <y> can be pressed in order to zoom only into y-direction.
  - When the right mouse button is being pressed, shifting can be performed.
    - Press the right mouse button at an axis (e.g. y-axis, t-axis or x-axis) and move the mouse cursor in order to move the scaling into the direction of the mouse move. t-axes of running online data visualizations can not be shifted because they always display the current time at their right border. In order to shift the scaling of a t-axis which contains running online data, the visualization must be paused via the toolbar first.
    - Press the right mouse button within the curve area and move the mouse cursor in order to move the scaling of all present axes into the direction of the mouse move. <Shift> or <x> can be pressed in order to move only into x-direction, <Ctrl> or <y> can be pressed in order to move only into y-direction. t-axes of running online data visualizations can not be moved because they always display the current time at their right border.

- Via the tool bar in the right corner of each **Monitoring Chart**, basic functionalities are available:
  - the cursors can be turned on/off
  - the last zooming, shifting and scrolling operations can be undone/redone
  - the visualization can be paused/continued
- Additional **Monitoring Charts** can be dragged from the **MTS Explorer** and can be placed into the opened Monitoring View. Data from the **MDS Explorer** can be put into the additional **Monitoring Charts** like they can be put into the first opened **Monitoring Chart** and the present **Monitoring Charts** can be arranged within their Monitoring View like the other windows can be arranged within ***X-Tools***.
- The Save and Save As… buttons are used in order to save the configured Monitoring View with all of its settings. All saved Monitoring Views are shown by the **MTS Explorer** ("Main -> Servers -> *YourServer* -> *YourLocation* -> Monitoring View Files" branch) and can be accessed (e.g. opened, …) from there.
- For a more detailed description of all available functionalities of **Monitoring View Editors** and **Monitoring Charts**, refer to the reference manual of the "Monitoring System", chapter "Monitoring View Editors" and chapter "Monitoring Charts".

Try it out

➔ Create a new Monitoring View.

➔ Drag some online data from the **MDS Explorer** into the Monitoring View in order to visualize the data curves.

➔ Pause the visualization, turn the cursors on and take a look onto the displayed measurement values. Continue the visualization, zoom into the visualization, shrink and stretch the axes scaling and use the undo/redo buttons. Repeat these operations in random order in order to adjust the visualization to the needs of the current application.

➔ Open two additional **Monitoring Charts** within the Monitoring View and start to visualize data also with the two additional **Monitoring Charts**.

➔ Save the Monitoring View. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to "Hello Monitoring View" and press OK in order to save it to the disk.

The new Monitoring View now is displayed below the "Main -> Servers -> *YourServer* -> *YourLocation* -> Monitoring View Files" branch of the **MTS Explorer**.

➔ Take a look onto the Monitoring Views which are provided via the example location (see point 6.5) in order to see various application examples about the capabilities and usage of Monitoring Views.

Ready

At this point you know how to create a new Monitoring View which can be used in order to visualize any of the data which is available from the **MDS Explorer**.

6.8.3 Opening of existing Monitoring Views

Open the MTS Explorer

- Open the **Monitoring System** and ensure that the **MTS Explorer** is displayed at the left side of the **Monitoring System** workspace.

Functionality

- Within the **MTS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Monitoring View Files” branch.
- Drag one of the below Monitoring Views with the left mouse into the empty space at the right side of the **MTS Explorer**. A **Monitoring View Editor** of proper type is opened automatically and displays the dropped Monitoring View.

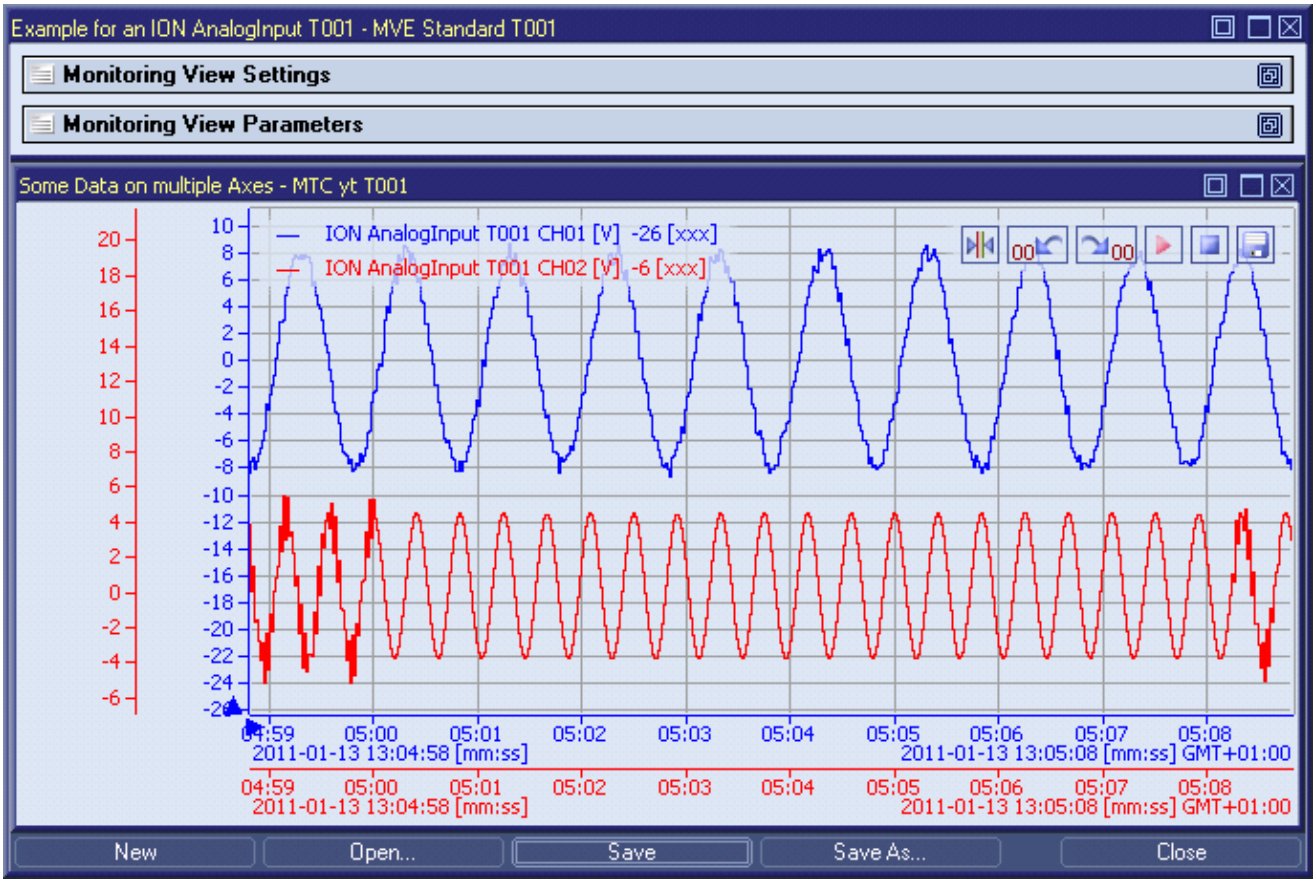

Example for an ION AnalogInput T001 - MVE Standard T001
Monitoring View Settings
Monitoring View Parameters
Some Data on multiple Axes - MTC yt T001
ION AnalogInput T001 CH01 [V] -26 [xxx]
ION AnalogInput T001 CH02 [V] -6 [xxx]
2011-01-13 13:04:58 [mm:ss]
2011-01-13 13:05:08 [mm:ss] GMT+01:00
New
Open...
Save
Save As...
Close

*Figure 27: **MVE Standard T001** with an opened Monitoring View*

- The opened Monitoring View can be edited now and can be used for any visualization of data. After editing, the Save and Save As… buttons are used in order to save the edited Monitoring View with all of its settings.

Try it out

➔ Open an existing Monitoring View.

➔ Add additional data to the present **Monitoring Charts** and change the scaling of some of the present axes.

➔ Save the Monitoring View via the Save button.

The modified Monitoring View is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Monitoring View Files” branch of the **MTS Explorer**.

Ready

At this point you know how to open and modify an existing Monitoring View.

6.8.4 Starting and Stopping of Monitoring Views

Monitoring Views can not be started or stopped. Instead, each Monitoring View starts to visualize the configured data according to the configured settings immediately when it is being opened and stops the visualization immediately when it is being closed.

6.9 Storage of online Data via Storage Profiles

6.9.1 General Description

In this chapter you will learn how to store online data from the **MDS Explorer** with the **Storage System** via Storage Profiles. A Storage Profile for online data contains all of the information which is needed in order to perform the storage of online data to a storage location. This information includes, among others, the names of the to-be-stored data, paths, trigger settings and clean-up settings. Multiple Storage Profiles can be started and running simultaneously at any time.

The online data which is being stored by a running Storage Profile can be loaded as offline data afterwards via an **Offline Data Loader**.

Examples of Storage Profiles are found within the example Configuration File location which is being delivered together with ***X-Tools***. See also point 6.15 for information about how to store offline data via Storage Profiles.

This chapter contains the following topics:

- **Creation of a new Storage Profile for online Data**
- **Opening of existing Storage Profiles**
- **Starting and Stopping of Storage Profiles**

6.9.2 Creation of a new Storage Profile for online Data

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch of the **MDS Explorer**.

Functionality

- Method 1 for the creation of a new Storage Profile:
  - Within the **STS Explorer**, expand the “Main -> Editors -> Storage Profile Editors” branch.
  - Drag the **SPE Binary T001** with the left mouse into the empty space at the right side of the **STS Explorer**. An empty **SPE Binary T001** is opened automatically.
  - Press the New button in order to create a new Storage Profile. The new Storage Profile is displayed by the **SPE Binary T001** with default values.
  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment. It can be used for the creation and configuration of Storage Profiles for cases where the target ***X-Tools Server*** is not present at the moment.
- Method 2 for the creation of a new Storage Profile:
  - Within the **STS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> Storage Process Modules” branch.
  - Click onto the present STE SPM Binary T001 with the right mouse button in order to open the context menu.
  - In order to create a new Storage Profile for the storage format “Binary T001”, choose New Storage Profile from the context menu.
  - This method works only in case the target ***X-Tools Server*** is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new Storage Profile.

- Method 3 for the creation of a new Storage Profile:
  - Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch.
  - Drag&Drop the desired target data into the empty **Storage System**. The **SPE Binary T001** is opened automatically, a new Storage Profile is being initialized with default values and the dropped data is added to Data table. As online data is being dropped, the Time Domain cell within the Storage Profile Parameters table is being set to “Online Data with absolute Timestamps” automatically.
  - This method works only in case the target ***X-Tools Server*** and the target data are present but it is more convenient and faster as it automatically opens all windows.
- The following screenshot shows an example of a new Storage Profile.

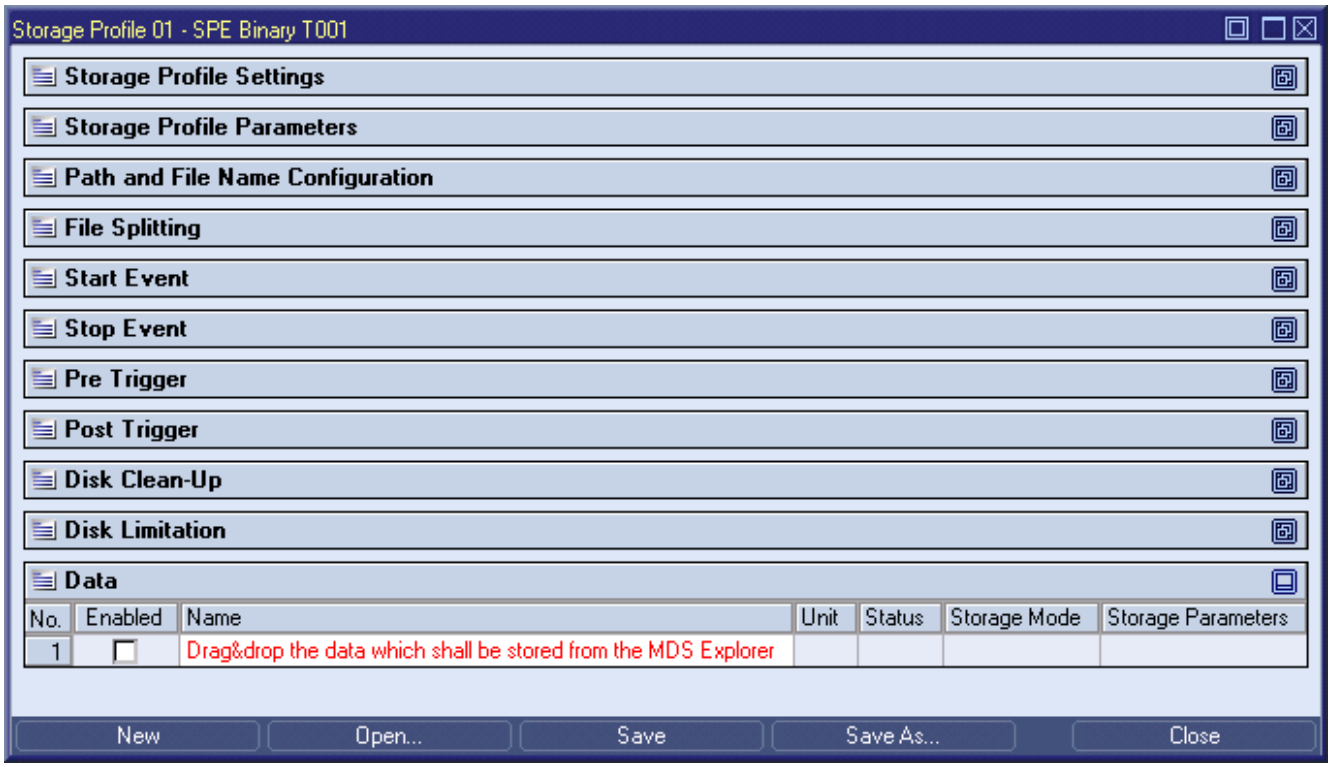


*Figure 28: **SPE Binary T001** with a new Storage Profile*

- The Storage Profile Settings table at the top of the **SPE Binary T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- Within the Storage Profile Parameters table, the time domain of the source data and the target location for the to-be-created offline data files are specified. Any of the already defined offline data file locations can be chosen from the combo box of the Data Storage Location cell. The “…” entry from the Data Storage Location cell of the Storage Profile Parameters table can be used in order to define a new offline data file location.
- Within the Subdirectory Creation table, the granularity of the levels of the to-be-created directories can be configured.
- Within the File Splitting table, the mode and the parameters of the file splitting can be configured.
- Within the Start Event table, the mode and the parameters of the start event can be configured. The storing of the configured data can be started in dependence to the current value of a data, date based or timer based.
- Within the Stop Event table, the mode and the parameters of the stop event can be configured. The storing of the configured data can be stopped in dependence to the current value of a data, date based or timer based.
- Within the Pre Trigger table, the mode and the parameters of the pre trigger can be configured.
- Within the Post Trigger table, the mode and the parameters of the post trigger can be configured.

- Within the Disk Clean-Up table, the mode and the parameters of the disk clean-up can be configured. The clean-up of files can be based on the age of files, it can limit the used memory or it can assure some remaining memory.
- Within the Disk Limitation table, the mode and the parameters of the disk limitation can be configured. In case the disk limitation is enabled, the Storage Profile will leave the specified absolute or relative amount of memory free at the hard disk.
- Within the Data table, the to-be-stored data are listed. In order to add an additional data which shall be stored, simple enter the name of the data into the Name cell of the last line. Alternatively, the to-be-stored data also can be dragged from the **MDS Explorer** into the Name cell of the last line.
- The Storage Mode and Storage Parameters cells of each to-be-stored data can be used in order to configure the intervals in which the data shall be stored.
- The Save and Save As… buttons are used in order to save the configured Storage Profile with all of its settings. All saved Storage Profiles are shown by the **STS Explorer** ("Main -> Servers -> *YourServer* -> *YourLocation* -> Storage Profile Files" branch) and can be accessed (e.g. started, stopped, opened, …) from there.
- For a more detailed description of all available functionalities of the **SPE Binary T001**, refer to the reference manual of the "Storage System", chapter "SPE Binary T001".

Try it out

➔ Create a new Storage Profile.

➔ Specify the desired data storage location.

➔ Change the file splitting interval to 1 minute.

➔ Add some data names to the Data table. Specify different storage modes for each of the to-be-stored data.

➔ Save the Storage Profile. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to "Hello Storage Profile for online Data" and press OK in order to save it to the disk.

The new Storage Profile now is displayed below the "Main -> Servers -> *YourServer* -> *YourLocation* -> Storage Profile Files" branch of the **STS Explorer**.

➔ Take a look onto the Storage Profiles which are provided via the example location (see point 6.5) in order to see various application examples about the capabilities and usage of Storage Profiles.

Ready

At this point you know how to create a new Storage Profile which can be used in order to store online data to the disk.

6.9.3 Opening of existing Storage Profiles

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.

Functionality

- Within the **STS Explorer**, expand the "Main -> Servers -> *YourServer* -> *YourLocation* -> Storage Profile Files" branch.
- Drag one of the below Storage Profiles with the left mouse into the empty space at the right side of the **STS Explorer**. A **Storage Profile Editor** of proper type is opened automatically and displays the dropped Storage Profile.

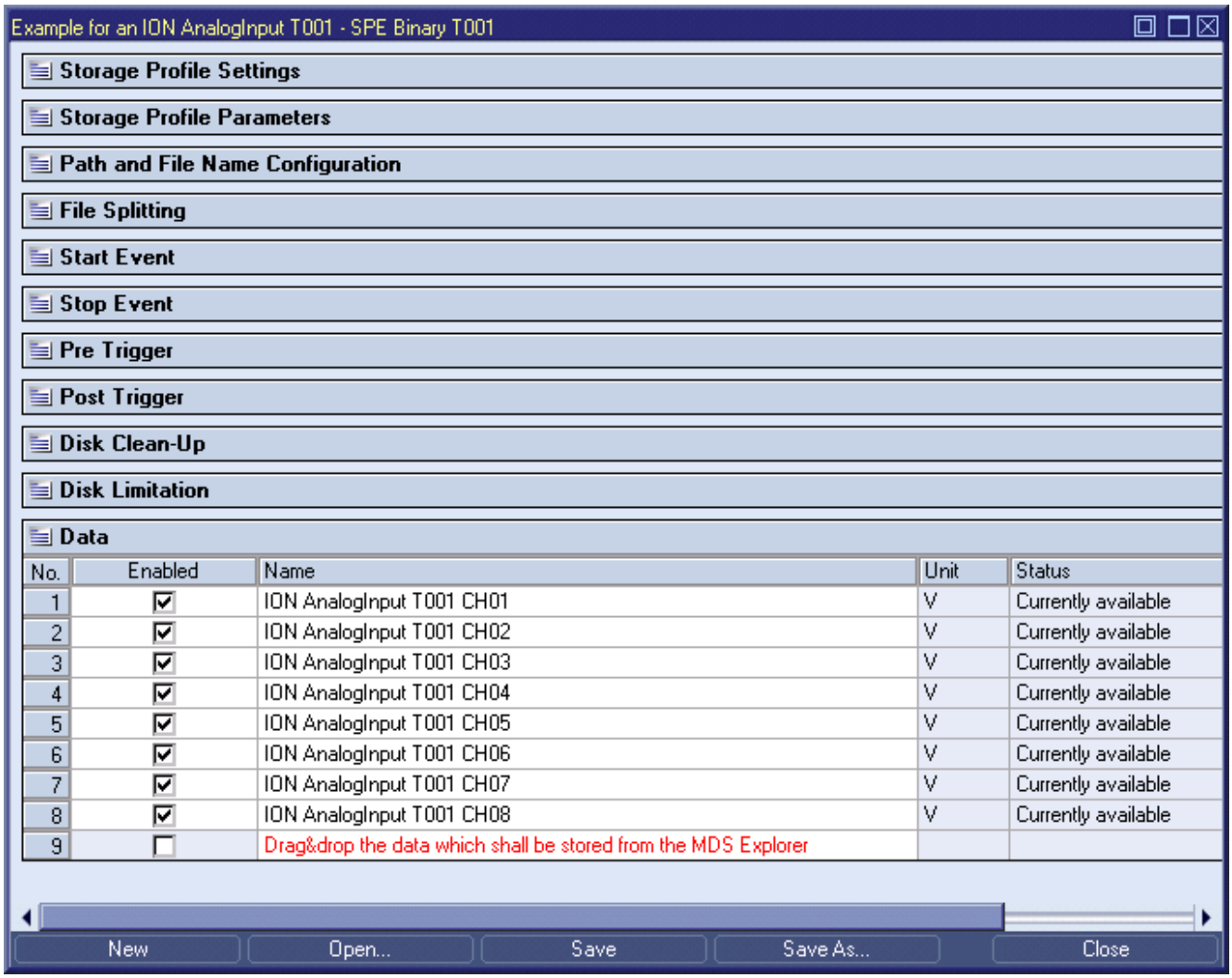

Example for an ION AnalogInput T001 - SPE Binary T001
Storage Profile Settings
Storage Profile Parameters
Path and File Name Configuration
File Splitting
Start Event
Stop Event
Pre Trigger
Post Trigger
Disk Clean-Up
Disk Limitation
Data
No.
Enabled
Name
Unit
Status
1 ION AnalogInput T001 CH01 V Currently available
2 ION AnalogInput T001 CH02 V Currently available
3 ION AnalogInput T001 CH03 V Currently available
4 ION AnalogInput T001 CH04 V Currently available
5 ION AnalogInput T001 CH05 V Currently available
6 ION AnalogInput T001 CH06 V Currently available
7 ION AnalogInput T001 CH07 V Currently available
8 ION AnalogInput T001 CH08 V Currently available
9 Drag&drop the data which shall be stored from the MDS Explorer
New
Open...
Save
Save As...
Close

*Figure 29: **SPE Binary T001** with an opened Storage Profile*

- The opened Storage Profile can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited Storage Profile with all of its settings.

Try it out

➔ Open an existing Storage Profile.

➔ Enter another storage location and change the Profile Description within the Storage Profile Settings table.

➔ Save the Storage Profile via the Save button.

The modified Storage Profile is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Storage Profile Files” branch of the **STS Explorer**.

Ready

At this point you know how to open and modify an existing Storage Profile.

6.9.4 Starting and Stopping of Storage Profiles

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch of the **MDS Explorer**.

Functionality

- Within the **STS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Storage Profile Files” branch.
- Click onto any of the present Storage Profiles with the right mouse button in order to open the context menu.
- In order to start the currently selected Storage Profile, choose Start from the context menu.
  - Multiple Storage Profiles can be running simultaneously and without any dependencies between each other.
- The stored data of each started Storage Profile is found at the specified data storage location. From there, the offline data can be loaded at any later point in time via the according **Offline Data Loader**.
- In order to stop a currently running Storage Profile, call the context menu of this Storage Profile and choose Stop.
- There are several conditions in which a Storage Profile can not run after it has been started, e.g. in case the source data is not available at the moment or in case the configured data storage location can not be accessed. When a Storage Profile can not run after it has been started, it is being suspended and automatically resumed as soon as the error condition has disappeared.
- In case at least one Storage Profile has been started by the user, the status bar of the ***X-Tools Client*** displays the current status of Storage Profiles via the according, colored icon. The tool tip of this icon provides detailed information about the number of currently started and suspended Storage Profiles.
  - Note that the status bar always displays the status of exactly one ***X-Tools Server***, thus also the icon for the current status of Storage Profiles displays the status of only the currently chosen ***X-Tools Server***.

Try it out

➔ Start an existing Storage Profile. Ensure that the configured, to-be-stored data is available within the **MDS Explorer**.

The data which has been configured within the started Storage Profile is being stored to the specified data storage location.

➔ Stop the currently running Storage Profile.

All of the currently opened offline data files are being closed.

Ready

At this point you know how to start and stop existing Storage Profiles in order to start and stop the storing of online data into offline data files.

6.10 Loading of Data and Loading Profiles

6.10.1 General Description

In this chapter you will learn how to load offline data from the disk with the **Storage System** with a simple method and via Loading Profiles. A Loading Profile contains all of the information which is needed in order to load data from a storage location. This information includes, among others, the names of the to-be-loaded data, the desired loading intervals and the desired period of time.

Examples of Loading Profiles are found within the example Configuration File location which is being delivered together with ***X-Tools***.

This chapter contains the following topics:

- **Simple Loading of Offline Data**
- **Creation of a new Loading Profile**
- **Opening of existing Loading Profiles**
- **Starting and Stopping of Loading Profiles**
- **Advanced Loading of Offline Data**

6.10.2 Simple Loading of Offline Data

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.

Functionality

- Method 1 for starting of the simple loading of offline data:
  - Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer*” branch.
  - Click onto the present Offline Data branch with the right mouse button in order to open the context menu.
  - In order to open the **ODL Standard T001**, choose Load Offline Data from the context menu. The **ODL Standard T001** is opened and its Offline Data Files tree automatically displays the available offline data file locations of the ***X-Tools Server*** from which the context menu has been called.
- Method 2 for starting of the simple loading of offline data:
  - Within the **STS Explorer**, expand the “Main -> Editors -> Offline Data Loaders” branch.
  - Drag the **ODL Standard T001** with the left mouse into the empty space at the right side of the **STS Explorer**. An empty **ODL Standard T001** is opened automatically.
  - Drag the ***X-Tools Server*** whose data shall be loaded into the empty **ODL Standard T001**. The Offline Data Files tree automatically displays the available offline data file locations of the dropped ***X-Tools Server***.
- Method 3 for starting of the simple loading of offline data:
  - Within the **STS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> Storage Process Modules” branch.
  - Click onto any of the present Storage Process Modules with the right mouse button in order to open the context menu.
  - In order to open the **ODL Standard T001**, choose Load Offline Data from the context menu. The **ODL Standard T001** is opened and its Offline Data Files tree automatically displays the available offline data file locations of the ***X-Tools Server*** from which the context menu has been called.

- The following screenshot shows an example of an opened **ODL Standard T001**.

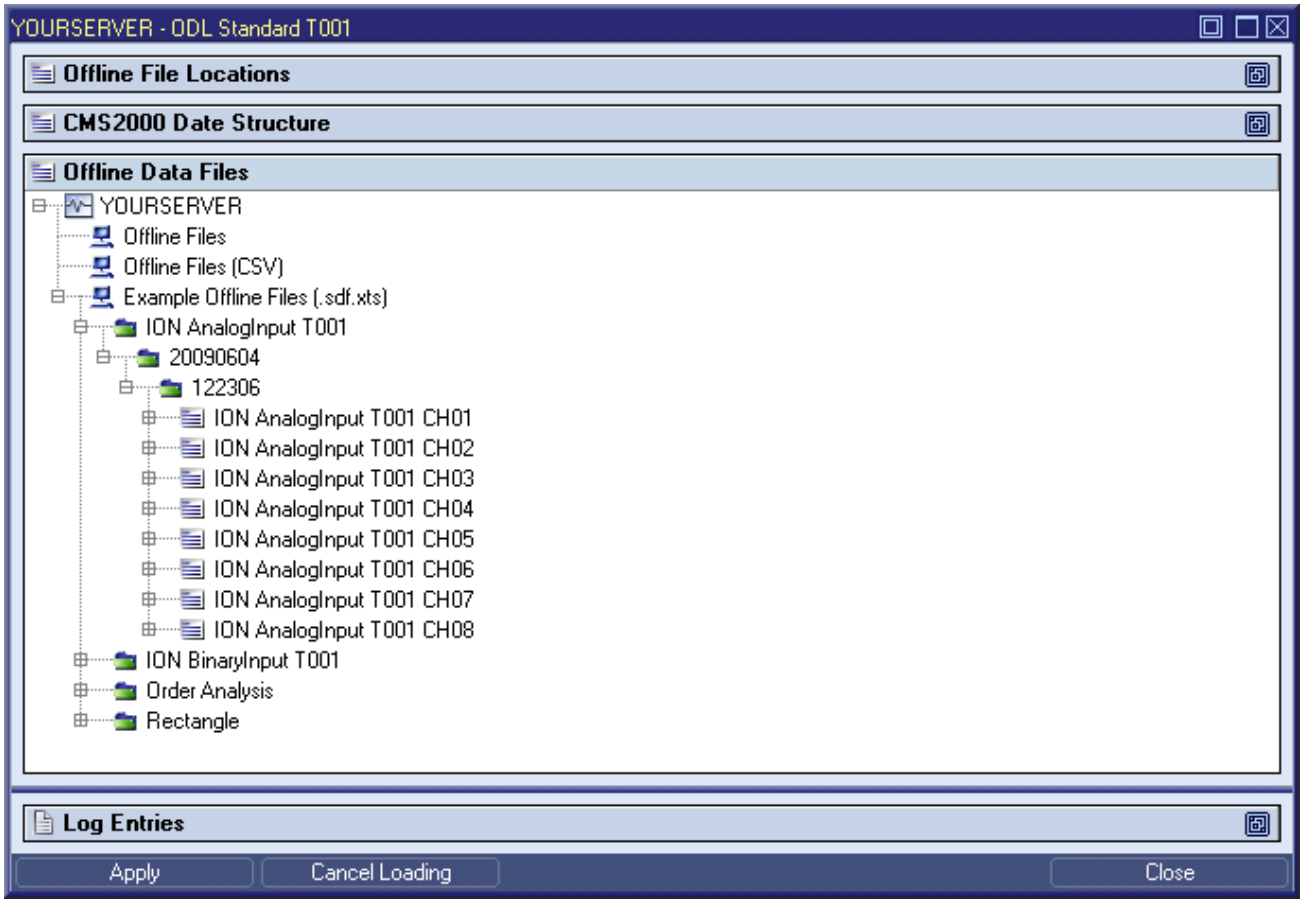


*Figure 30: **ODL Standard T001** with some Offline Data Files*

- The Offline File Locations table at the top of the **ODL Standard T001** displays the symbolic names and absolute paths of the available offline data file locations.
- The CM2000 Date Structure table allows to specify the depth of the date structure which shall be shown for CMS2000 files.
- The Offline Data Files tree displays the available offline data file locations together with all of the offline data files and directories which are present below the offline data file location.
- The contents of the Offline Data Files tree can be browsed for the to-be-loaded offline data by expanding and collapsing the displayed directories. In order to load some offline data, the context menu can be called above any file or directory. It also is possible to select multiple files or multiple directories simultaneously and to load their information at once.
  - The Append context menu item can be used in order to append the currently selected files/directories to the already loaded parts of offline data.
  - The Overwrite context menu item can be used in order to overwrite the already loaded parts of offline data with the currently selected files/directories.
  - The Export context menu item can be used in order to export the currently selected files/directories. The exported files are found in the default offline data file location.
  - Additional information about the Advanced Append…, Advanced Overwrite… and Advanced Export… context menu items is found later in this document (see point 6.10.6).
- The **ODL Standard T001** supports to add and to use offline file locations of different type. In addition to ***X-Tools***-generated offline data files (*.sdf.xts) also the database files from the CMS2000 system (*.cmsdb), WAVE files (*.wav) and CSV files (*.csv) can be loaded.
- While the appending, overwriting or exporting of the offline data is going on, the **Loading Progress** or **Export Progress** dialog is being displayed. The Cancel button can be used in order to cancel any ongoing loading or export operation.

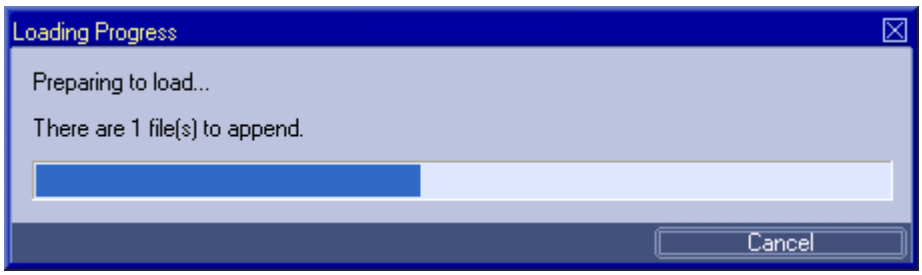


Figure 31: ***Loading Progress*** *Dialog*

- Each loaded offline data is displayed by the **MDS Explorer** below its “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branch. From there, it can be dragged into the other systems of ***X-Tools*** for further analysis and visualization tasks.
- For a more detailed description of all available functionalities of the **ODL Standard T001**, refer to the reference manual of the “Storage System”, chapter “ODL Standard T001”.

Try it out

➔ Open the **ODL Standard T001** for the local ***X-Tools Server***.

➔ Browse your offline data file locations for already stored offline data files.

➔ Load some offline data files and append additional periods of time to the first loaded data afterwards.

The loaded offline data is being displayed within the **MDS Explorer**.

➔ Overwrite the already loaded offline data with another period of time of the data.

The information within the **MDS Explorer** is being updated accordingly.

➔ Export some of the present offline data files.

According export files are being created and can be opened with other applications.

Ready

At this point you know how to use the simple mechanisms of the **ODL Standard T001** in order to load offline data files which have been stored to disk previously.

6.10.3 Creation of a new Loading Profile

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch of the **MDS Explorer**.

Functionality

- Method 1 for the creation of a new Loading Profile:
  - Within the **STS Explorer**, expand the “Main -> Editors -> Loading Profile Editors” branch.
  - Drag the **LPE Standard T001** with the left mouse into the empty space at the right side of the **STS Explorer**. An empty **LPE Standard T001** is opened automatically.
  - Press the New button in order to create a new Loading Profile. The new Loading Profile is displayed by the **LPE Standard T001** with default values.
  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment. It can be used for the creation and configuration of Loading Profiles for cases where the target ***X-Tools Server*** is not present at the moment.

- The following screenshot shows an example of a new Loading Profile.

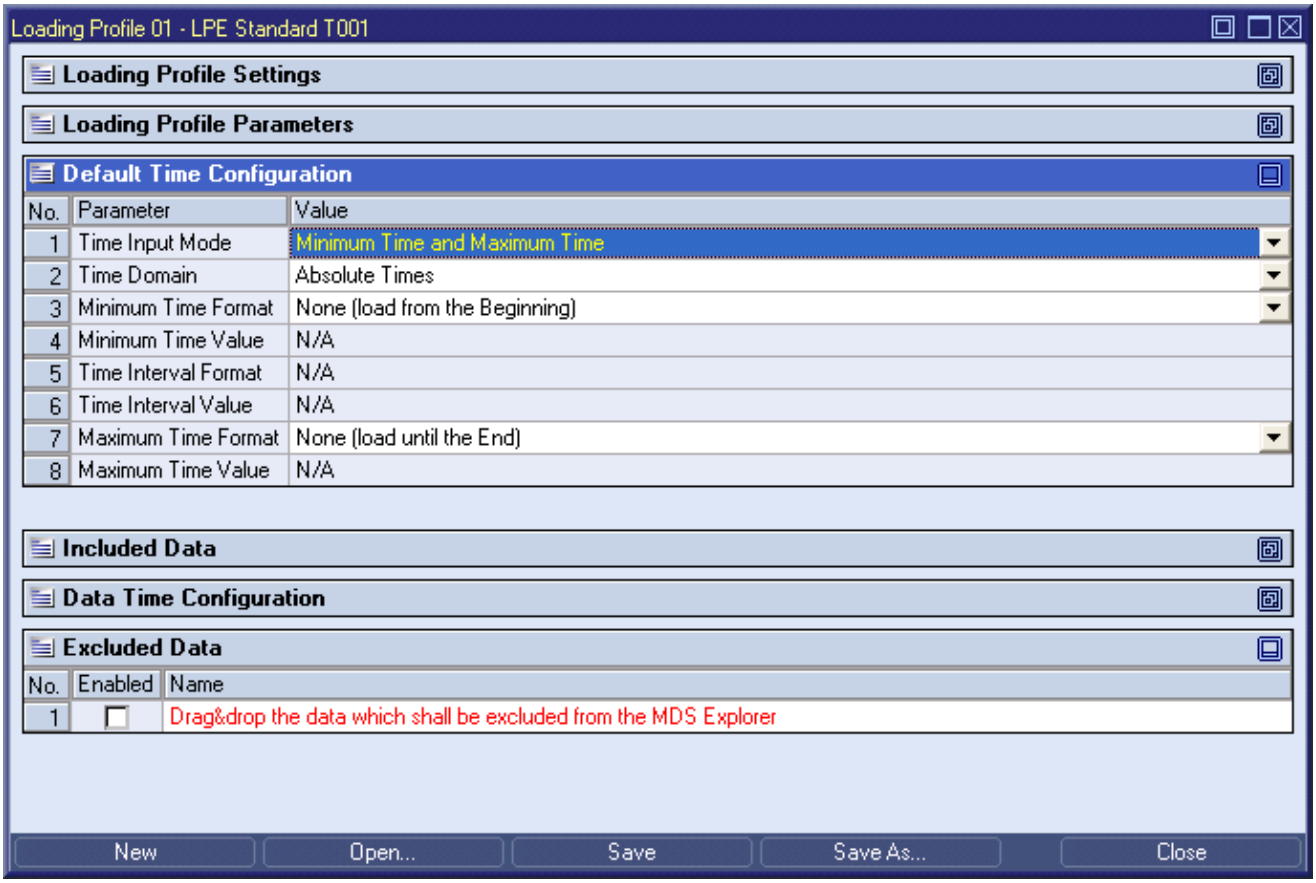

Loading Profile 01 - LPE Standard T001
Loading Profile Settings
Loading Profile Parameters
Default Time Configuration
No. Parameter Value
1 Time Input Mode Minimum Time and Maximum Time
2 Time Domain Absolute Times
3 Minimum Time Format None (load from the Beginning)
4 Minimum Time Value N/A
5 Time Interval Format N/A
6 Time Interval Value N/A
7 Maximum Time Format None (load until the End)
8 Maximum Time Value N/A
Included Data
Data Time Configuration
Excluded Data
No. Enabled Name
1 Drag&drop the data which shall be excluded from the MDS Explorer
New
Open...
Save
Save As...
Close

*Figure 32: **LPE Standard T001** with a new Loading Profile*

- The Loading Profile Settings table at the top of the **LPE Standard T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- Within the Loading Profile Parameters table, the to-be-used configuration can be configured.
  - Choose “Use included Data List” in order to use the configuration from the Included Data table.
  - Choose “Use excluded Data List” in order to use the configuration from the Excluded Data table.
  - Choose “Use default Time” in order to apply the settings from the Default Time Configuration table to all of the data which are being loaded.
  - Choose “Use individual Times” in order to allow individual configuration of the to-be-loaded time interval for each data from the Included Data table.
- Within the Default Time Configuration table, default time interval for loading of offline data can be configured.
- Within the Included Data table, the data which shall be loaded are listed. Data which is present within the directory that is being loaded but is not present within the Included Data table is not being loaded in case “Use included Data List” is being chosen.
  - In order to add an additional data which shall be loaded, simple enter the name of the data into the Name cell of the last line. Alternatively, the to-be-loaded data also can be dragged from the **MDS Explorer** into the Name cell of the last line.
  - The Loading Mode and Loading Parameters cells of each to-be-loaded data can be used in order to configure the intervals in which the data shall be loaded.
  - When an entry of the Included Data table is selected, the Data Time Configuration table displays its current time configuration and allows editing it.
- Within the Excluded Data table, the data which shall not be loaded are listed. Data which is present within the Excluded Data table is not being loaded in case “Use excluded Data List” is being chosen.

- The Save and Save As… buttons are used in order to save the configured Loading Profile with all of its settings. All saved Loading Profiles are shown by the **STS Explorer** (“Main -> Servers -> *YourServer* -> *YourLocation* -> Loading Profile Files” branch) and can be accessed (e.g. opened, …) from there.
- For a more detailed description of all available functionalities of the **LPE Standard T001**, refer to the reference manual of the “Storage System”, chapter “LPE Standard T001”.

Try it out

➔ Create a new Loading Profile.

➔ Specify the data which shall be included during the loading.

➔ Change the loading mode of some data so that only 1 value per second is being loaded.

➔ Change the time configuration of some data so that only 10 seconds of the data are being loaded.

➔ Save the Loading Profile. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to “Hello Loading Profile” and press OK in order to save it to the disk.

The new Loading Profile now is displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Loading Profile Files” branch of the **STS Explorer**.

➔ Take a look onto the Loading Profiles which are provided via the example location (see point 6.5) in order to see various application examples about the capabilities and usage of Loading Profiles.

Ready

At this point you know how to create a new Loading Profile which can be used in order to load offline data from the disk.

6.10.4 Opening of existing Loading Profiles

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.

Functionality

- Within the **STS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Loading Profile Files” branch.
- Drag one of the below Loading Profiles with the left mouse into the empty space at the right side of the **STS Explorer**. A **Loading Profile Editor** of proper type is opened automatically and displays the dropped Loading Profile.

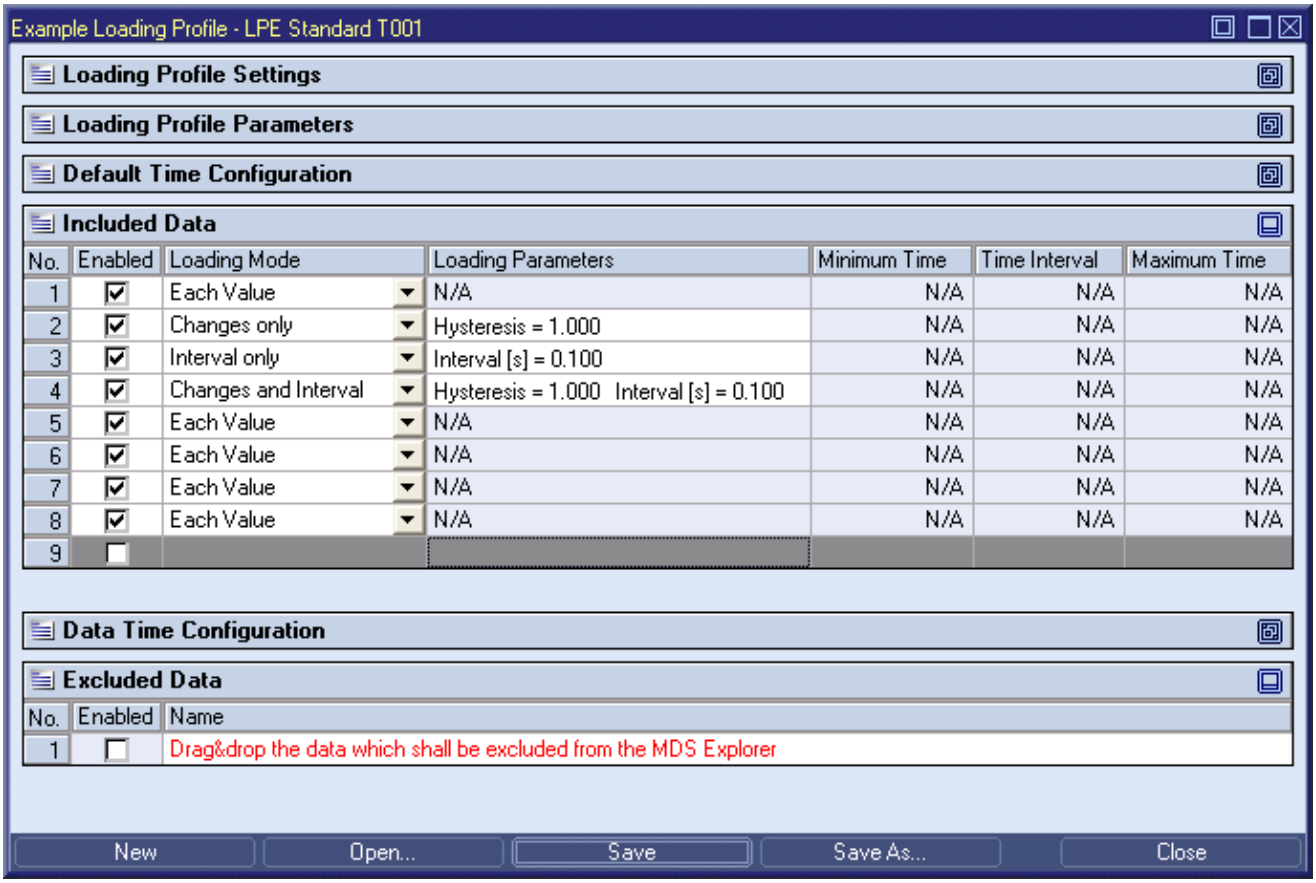

Example Loading Profile - LPE Standard T001
Loading Profile Settings
Loading Profile Parameters
Default Time Configuration
Included Data
No. Enabled Loading Mode Loading Parameters Minimum Time Time Interval Maximum Time
1 Each Value N/A N/A N/A N/A
2 Changes only Hysteresis = 1.000 N/A N/A N/A
3 Interval only Interval [s] = 0.100 N/A N/A N/A
4 Changes and Interval Hysteresis = 1.000 Interval [s] = 0.100 N/A N/A N/A
5 Each Value N/A N/A N/A N/A
6 Each Value N/A N/A N/A N/A
7 Each Value N/A N/A N/A N/A
8 Each Value N/A N/A N/A N/A
9
Data Time Configuration
Excluded Data
No. Enabled Name
1 Drag&drop the data which shall be excluded from the MDS Explorer
New
Open...
Save
Save As...
Close

*Figure 33: **LPE Standard T001** with an opened Loading Profile*

- The opened Loading Profile can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited Loading Profile with all of its settings.

Try it out

➔ Open an existing Loading Profile.

➔ Enter some additional data which shall be loaded and change the Profile Description within the Loading Profile Settings table.

➔ Save the Loading Profile via the Save button.

The modified Loading Profile is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Loading Profile Files” branch of the **STS Explorer**.

Ready

At this point you know how to open and modify an existing Loading Profile.

6.10.5 Starting and Stopping of Loading Profiles

Loading Profiles can not be started or stopped. Instead, Loading Profiles are being used by **Offline Data Loaders** for their advanced append, overwrite and export operations.

6.10.6 Advanced Loading of Offline Data

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.

Functionality

- Like the simple loading of offline data, also the advanced loading of offline data uses the **ODL Standard T001** for the actual loading. Opening of the **ODL Standard T001** is identical no matter whether the simple or the advanced mechanisms for loading of offline data are going to be used.
- Method 1 for starting of the advanced loading of offline data:
  - Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer*” branch.
  - Click onto the present Offline Data branch with the right mouse button in order to open the context menu.
  - In order to open the **ODL Standard T001**, choose Load Offline Data from the context menu. The **ODL Standard T001** is opened and its Offline Data Files tree automatically displays the available offline data file locations of the ***X-Tools Server*** from which the context menu has been called.
- Method 2 for starting of the advanced loading of offline data:
  - Within the **STS Explorer**, expand the “Main -> Editors -> Offline Data Loaders” branch.
  - Drag the **ODL Standard T001** with the left mouse into the empty space at the right side of the **STS Explorer**. An empty **ODL Standard T001** is opened automatically.
  - Drag the ***X-Tools Server*** whose data shall be loaded into the empty **ODL Standard T001**. The Offline Data Files tree automatically displays the available offline data file locations of the dropped ***X-Tools Server***.
- Method 3 for starting of the advanced loading of offline data:
  - Within the **STS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> Storage Process Modules” branch.
  - Click onto the present STE SPM Binary T001 with the right mouse button in order to open the context menu.
  - In order to open the **SPE Binary T001**, choose Load Offline Data from the context menu. The **ODL Standard T001** is opened and its Offline Data Files tree automatically displays the available offline data file locations of the ***X-Tools Server*** from which the context menu has been called.

- The following screenshot shows an example of an opened **ODL Standard T001**.

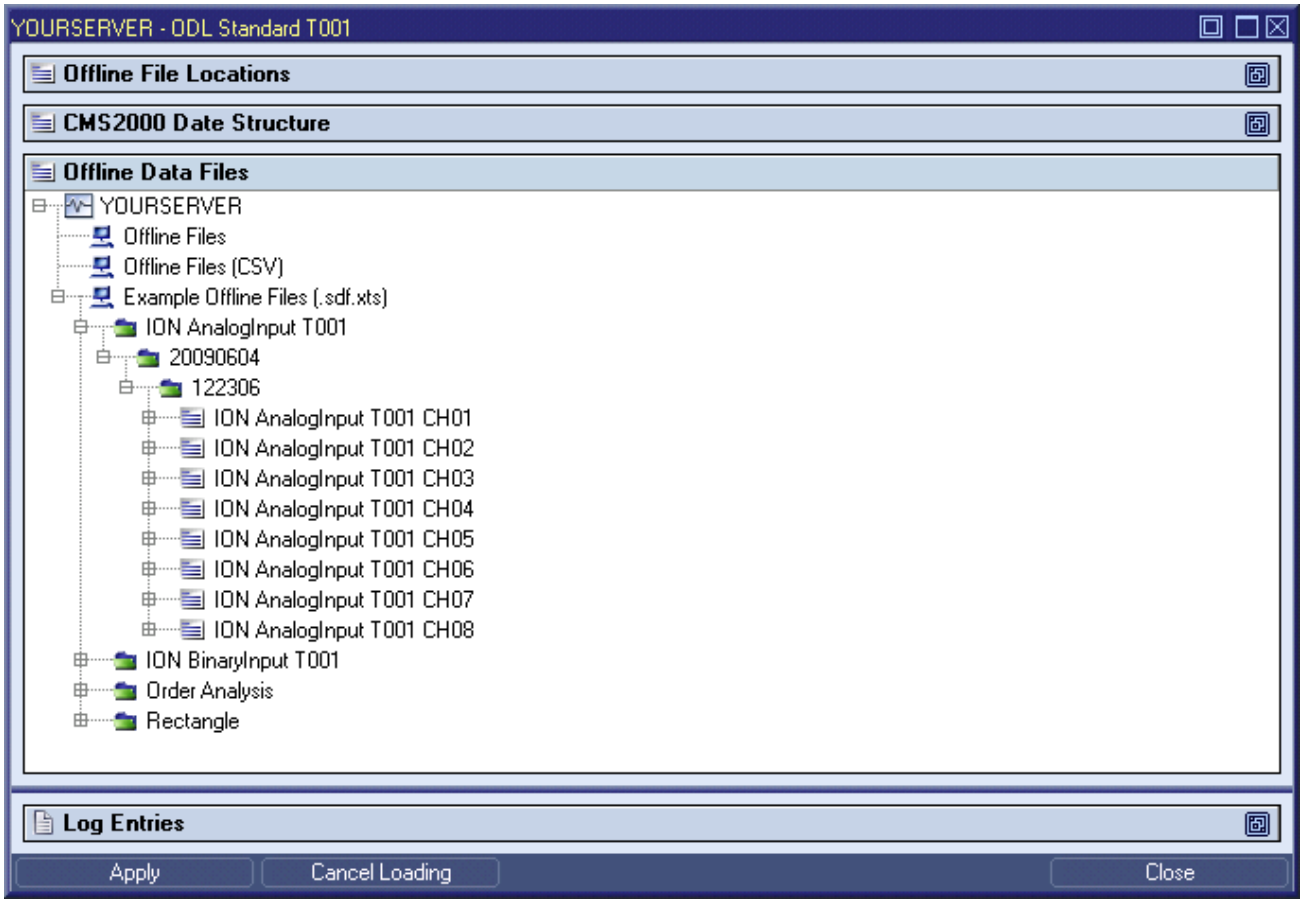


*Figure 34: **ODL Standard T001** with some Offline Data Files*

- The Offline File Locations table at the top of the **ODL Standard T001** displays the symbolic names and absolute paths of the available offline data file locations.
- The Offline Data Files tree displays the available offline data file locations together with all of the offline data files and directories which are present below the offline data file location.
- The contents of the Offline Data Files tree can be browsed for the to-be-loaded offline data by expanding and collapsing the displayed directories. In order to load some offline data, the context menu can be called above any file or directory. It also is possible to select multiple files or multiple directories simultaneously and to load their information at once.
  - The Advanced Append… context menu item can be used in order to call the **Advanced Append** dialog for the currently selected files/directories. Within the **Advanced Append** dialog, various settings can be provided which are applied when the actual appending of offline data is performed.
    - In case a Loading Profile is specified via the Used Loading Profile cell, the settings about to-be-included, to-be-excluded and/or desired time intervals and loading modes are read from the specified Loading Profile and applied during the loading of offline data.
    - Via the Loading Mode and Loading Parameters cells, intelligent data reduction can be applied when the offline data is being loaded.
    - Via the Time Input Mode cell and its below cells, the to-be-loaded time interval can be specified manually.
    - The OK button is used in order to start appending of the currently selected data according to the currently chosen settings.
  - The Advanced Overwrite… context menu item can be used in order to call the **Advanced Overwrite** dialog for the currently selected files/directories. Within the **Advanced Overwrite** dialog, various settings can be provided which are applied when the actual overwriting of offline data is performed.

        - In case a Loading Profile is specified via the Used Loading Profile cell, the settings about to-be-included, to-be-excluded and/or desired time intervals and loading modes are read from the specified Loading Profile and applied during the loading of offline data.
        - Via the Loading Mode and Loading Parameters cells, intelligent data reduction can be applied when the offline data is being loaded.
        - Via the Time Input Mode cell and its below cells, the to-be-loaded time interval can be specified manually.
        - The OK button is used in order to start overwriting with the currently selected data according to the currently chosen settings.
    - The Advanced Export… context menu item can be used in order to call the **Advanced Export** dialog for the currently selected files/directories. Within the **Advanced Export** dialog, various settings can be provided which are applied when the actual exporting of offline data is performed.
        - In case a Loading Profile is specified via the Used Loading Profile cell, the settings about to-be-included, to-be-excluded and/or desired time intervals and loading modes are read from the specified Loading Profile and applied during the exporting of offline data.
        - Via the Export Location and Export Path cells, the target location and path for the to-be-created export files can be specified.
        - Via the Export File Type, Timestamp Format and Decimal Precision cells, the format of the information within the to-be-created export files can be specified.
        - Via the Export Mode and Export Parameters cells, intelligent data reduction can be applied when the offline data is being exported.
        - Via the Time Input Mode cell and its below cells, the to-be-exported time interval can be specified manually.
        - The OK button is used in order to start exporting of the currently selected data according to the currently chosen settings.

- Each loaded offline data is displayed by the **MDS Explorer** below its “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branch. From there, it can be dragged into the other systems of ***X-Tools*** for further analysis and visualization tasks.
- For a more detailed description of all available functionalities of the **ODL Standard T001**, refer to the reference manual of the “Storage System”, chapter “ODL Standard T001”.

Try it out

➔ Open the **ODL Standard T001** for the local ***X-Tools Server***.

➔ Browse your offline data file locations for already stored offline data files.

➔ Load a time interval of only 10 seconds of some offline data files.

The loaded offline data is being displayed within the **MDS Explorer**.

➔ Overwrite the already loaded offline data with another period of time of the data. Use the loading mode in order to load only one value per second for each of the selected data.

The information within the **MDS Explorer** is being updated accordingly.

➔ Export some of the present offline data files. Use a Loading Profile in order to predefine which of the selected data shall be exported.

According export files are being created and can be opened with other applications.

Ready

At this point you know how to use the advanced mechanisms of the **ODL Standard T001** in order to load offline data files which have been stored to disk previously.

6.11 Offline File Locations

6.11.1 General Description

In this chapter you will learn how offline file locations can be used in order to maintain different directories which contain the to-be-loaded data files of interest.

This chapter contains the following topics:

- **Adding of an Offline File Location**
- **Removing of an Offline File Location**

6.11.2 Adding of an Offline File Location

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.

Functionality

- Method 1 for adding of an offline file location:
  - Open a Storage Profile within the **SPE Binary T001**.
  - Within the Storage Profile Parameters table, open the Data Storage Location combo box.
  - Choose the "…" entry in order to open the **Add Location** dialog.
  - This method can be used in order to create offline file locations of type "X-Tools Offline Data Files (*.sdf.xts)".
- Method 2 for adding of an offline file location:
  - Open an **ODL Standard T001**.
  - Within the Offline Data Files tree, open the context menu of the root item.
  - Choose Add Location… in order to open the **Add Location** dialog.
  - This method can be used in order to create offline file locations of any supported type.
- All of the currently present offline file locations are listed within the Offline File Locations table of the **ODL Standard T001**.
- Each offline file location shows only the files which match its configured location type.
- The new location can not be created within the sub-directory of an existing location or within the parent-directory of an existing location.
- One example directory for each supported type of offline file is delivered together with ***X-Tools***. These directories are found below "..\X-Tools\Server\User\Example Offline Files" and can be used in order to add and use each of the supported types of offline file locations.

Try it out

➔ Add a new directory as additional offline file location.

All of the offline data files of matching type from the added location become visible within the Offline Data Files tree of the **ODL Standard T001**.

Ready

At this point you know how to add offline file locations to the ***X-Tools Server***.

6.11.3 Removing of an Offline File Location

Open the ODL Standard T001

- Open the **Storage System** and ensure that the **ODL Standard T001** is displayed within the **Storage System** workspace.

Functionality

- Expand the main branch of the Offline Data Files tree within the **ODL Standard T001**.
- In order to remove an existing offline file location, call the context menu for any of the present offline file locations within the Offline Data Files tree and choose Remove Location.
- After confirming of the upcoming dialog, the chosen offline file location is being removed from the Offline Data Files tree.

Try it out

➔ Remove any of the present offline file locations.

All of the directories and files from the specified location are being removed from the Offline Data Files tree of the **ODL Standard T001**.

Ready

At this point you know how to remove offline file locations from the ***X-Tools Server***.

6.12 Visualization of offline Data

6.12.1 General Description

In this chapter you will learn how to visualize offline data from the **MDS Explorer** with the **Monitoring System** using the most basic functionalities.

This chapter contains the following topics:

- **Visualization of offline Data**

6.12.2 Visualization of offline Data

Open the Monitoring System

- Open the **Monitoring System**.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” and “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branches of the **MDS Explorer**. At least some data names shall be available below both branches.

Functionality

- There are nearly no differences between the visualization of online and offline data. The major difference is that offline data can not be running, thus the Pause Visualization and Continue Visualization buttons do not have any effects on offline data.
- The creation and handling of Monitoring Views is identical for online data and offline data (see 6.8).
- Within each Monitoring View, online data and offline data can be visualized simultaneously.
- In order to visualize any of the currently available offline data, drag the desired data from the “Main -> Servers -> *YourServer* -> Offline Data” branch of the **MDS Explorer** into an empty **MVE Standard T001** or into an opened **Monitoring Chart**. Up to 32 data can be put into one **Monitoring Chart**.
- Each Monitoring View can be switched between visualization of online data and visualization of offline data. In order to switch a the visualization between online data and offline data, the following possibilities are available:
  - The time domain (online or offline) of each **Monitoring Chart** can be changed via its **Chart Options** dialog (call the context menu of the **Monitoring Chart** and choose Chart Options… in order to open the **Chart Options** dialog). Within this dialog, the Time Domain cell of the Chart Options table allows to choose any of the available time domains. After the time domain has been changed, the **Monitoring Chart** starts to display the data with the known names from the specified time domains. In case there is no data available from the newly chosen time domain and with the known name, an according icon is displayed by the legend of the **Monitoring Chart**.
  - The time domain for the whole Monitoring View can be changed via the Time Domain cell of the Monitoring View Parameters table of the **MVE Standard T001**. The chosen time domain from the Monitoring View Parameters table is applied to all of the opened **Monitoring Charts** of the **MVE Standard T001** immediately.

Try it out

➔ Create a new Monitoring View.

➔ Drag some offline data from the **MDS Explorer** into the Monitoring View in order to visualize the data curves.

➔ Turn the cursors on and take a look onto the displayed measurement values. Zoom into the visualization, shrink and stretch the axes scaling and use the undo/redo buttons. Repeat these operations in random order in order to adjust the visualization to the needs of the current application.

➔ Open two additional **Monitoring Charts** within the Monitoring View and start to visualize data also with the two additional **Monitoring Charts**.

➔ Add online data and offline data to one **Monitoring Chart** in order to visualize them in parallel.

➔ Use the Time Domain cells of the **Chart Options** dialog and of the **MVE Standard T001** in order to switch the time domains of a single **Monitoring Chart** or of all **Monitoring Charts** simultaneously.

Ready

At this point you know how to visualize offline data, how to switch the visualization between online data and offline data and how to visualize and use online and offline data in parallel.

6.13 Analyzing of Data via Analyzing Models

6.13.1 General Description

In this chapter you will learn how to analyze data from the **MDS Explorer** with the **Analyzing System** via Analyzing Models. An Analyzing Model contains all of the information which is needed in order to perform calculations based on the specified input data. This information includes the order of execution and the parameters of all to-be-called (Analyzing and User) functions. Multiple Analyzing Models can be started and running simultaneously at any time. In addition, the data which are defined as output data within a started Analyzing Model are available to all other modules of ***X-Tools*** via the **MDS Explorer**.

Examples of Analyzing Models are found within the example Configuration File location which is being delivered together with ***X-Tools***.

This chapter contains the following topics:

- **Creation of a new Analyzing Model**
- **Opening of existing Analyzing Models**
- **Starting and Stopping of Analyzing Models**

6.13.2 Creation of a new Analyzing Model

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” and “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branches of the **MDS Explorer**.

Functionality

- Method 1 for the creation of a new Analyzing Model:
  - Within the **ANS Explorer**, expand the “Main -> Editors -> Analyzing Model Editors” branch.
  - Drag the **AME Standard T001** with the left mouse into the empty space at the right side of the **ANS Explorer**. An empty **AME Standard T001** is opened automatically.
  - Press the New button in order to create a new Analyzing Model. The new Analyzing Model is displayed by the **AME Standard T001** with default values.
  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment. It can be used for the creation and configuration of Analyzing Models for cases where the target ***X-Tools Server*** is not present at the moment.
- Method 2 for the creation of a new Analyzing Model:
  - This method works only in case the **AME Standard T001** is defined as “Default Editor for Functions” within the **Global Options Editor** of the ***X-Tools Client***.
  - Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> Analyzing Functions” branch.
  - Expand also the “Arithmetic” branch and drag a below Analyzing Function (e.g. the “Abs()” Analyzing Function) into the empty **Analyzing System** workspace. An **AME Standard T001** with a default Analyzing Model which also contains the dropped Analyzing Function is opened automatically.
  - Instead of an Analyzing Function from the “Main -> Servers -> *YourServer* -> Libraries -> Analyzing Functions” branch, also a User Function from the Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch can be dragged into the empty **Analyzing System**.
  - This method works only in case the target ***X-Tools Server*** is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new Analyzing Model.

- Method 3 for the creation of a new Analyzing Model:
  - This method works only in case the **AME Standard T001** is defined as “Default Editor for Functions” within the **Global Options Editor** of the ***X-Tools Client***.
  - Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer* -> Online Data” or the “Main -> Servers -> *YourServer* -> Offline Data” branch.
  - Drag an online or offline data from the **MDS Explorer** into the empty **Analyzing System** workspace. An **AME Standard T001** with a default Analyzing Model which also contains the dropped data is opened automatically.
  - This method works only in case the target data is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new Analyzing Model.
- The following screenshot shows an example of a new Analyzing Model.

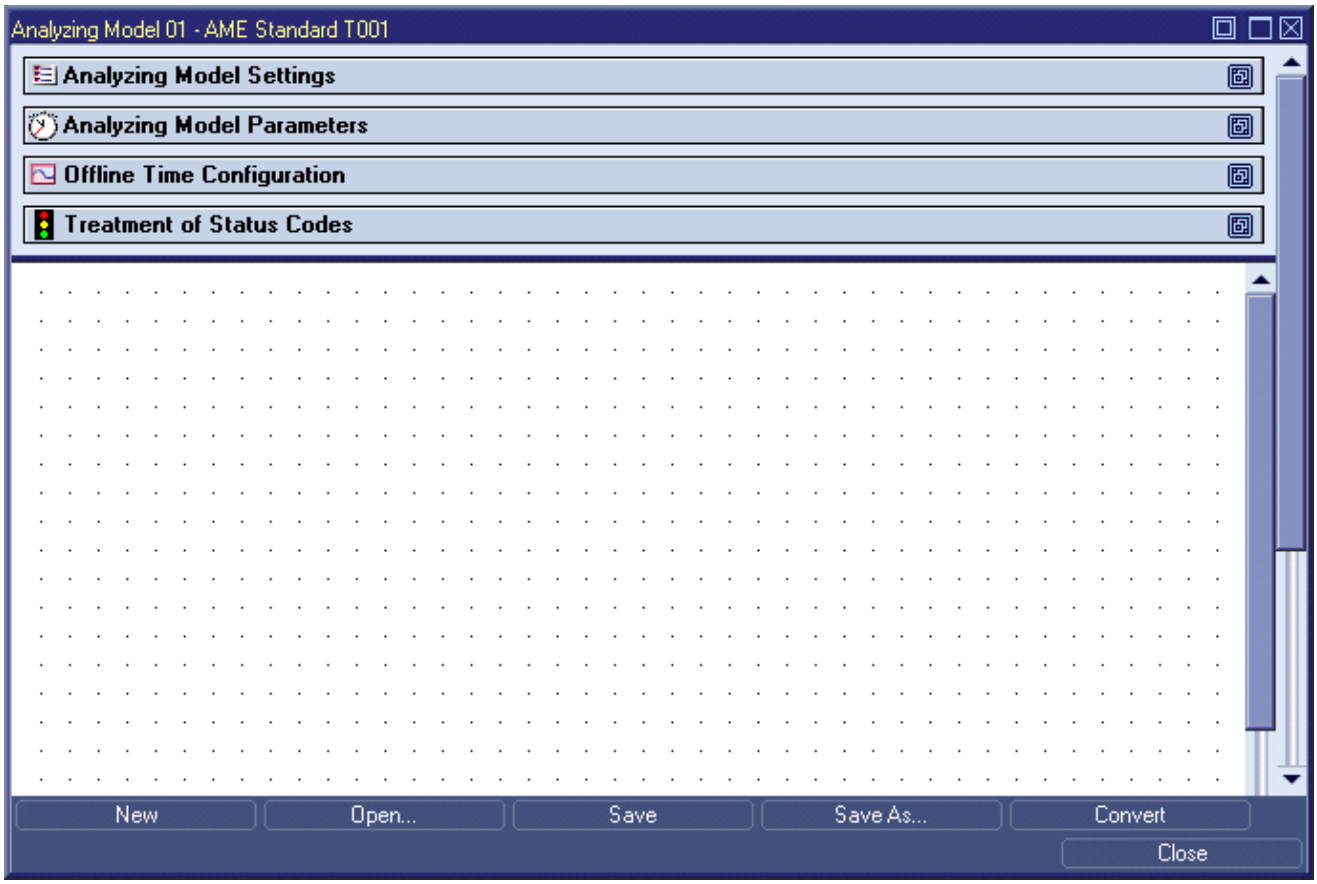


*Figure 35: **AME Standard T001** with a new Analyzing Model*

- The Analyzing Model Settings table at the top of the **AME Standard T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- Within the Analyzing Model Parameters table, the basic parameters for the execution of the Analyzing Model are being defined.
  - The default configuration of Analyzing Models typically allows executing the defined analysis without additional changes to the parameters.
  - For a detailed description of the meaning and usage of the parameters of Analyzing Models, refer to the reference manual of the “Analyzing System”, chapter “Analyzing Model Editors”.

- Within the Offline Time Configuration table, the time interval onto which the calculation shall be applied is being defined. The contents of this table can be configured only in case the output data class is specified as offline data.
  - The default configuration of Analyzing Models typically allows executing the defined analysis without additional changes to the offline time configuration.
  - For a detailed description of the meaning and usage of the offline time configuration of Analyzing Models, refer to the reference manual of the “Analyzing System”, chapter “Analyzing Model Editors”.
- Within the Treatment of Status Codes table, the handling of occurring status codes is being defined.
  - The default configuration of Analyzing Models typically allows executing the defined analysis without additional changes to the treatment of status codes.
  - For a detailed description of the meaning and usage of the treatment of status codes of Analyzing Models, refer to the reference manual of the “Analyzing System”, chapter “Analyzing Model Editors”.
- In order to add input data to the Analyzing Model, simply drag the desired data from the **MDS Explorer** into the Analyzing Model.
- In order to add Analyzing Functions to the Analyzing Model, simply drag the desired Analyzing Function from the **ANS Explorer** into the Analyzing Model.
- In order to add User Functions to the Analyzing Model, simply drag the desired User Function from the **ANS Explorer** into the Analyzing Model.
- (Analyzing and User) functions can be connected together via links. Each output of a function can be connected to one or more inputs of functions by clicking onto the output argument first and by clicking onto the input argument next. Each created link transports the output value from its source function to the input argument of its destination function.
- All of the currently present functions can be arranged within the Analyzing Model via Drag&Drop.
- A double-click onto a function opens the **Analyzing Function Properties** dialog which can be used in order to specify the values of the input arguments of the function.
- Via the context menu, functions can be cut/copied/pasted/removed and their visualization can be modified.
- An Analyzing Model can work either only on online data, only on offline data or on online and offline data simultaneously. The class of the resulting data (online or offline) can be configured via the Analyzing Model Parameters table.
- Analyzing Functions from the category “Input” are being used in order to get (online or offline) data from the system into the Analyzing Model. Analyzing Functions from the category “Output” are being used in order to put (online or offline) data from the Analyzing Model back to the system.
- The Save and Save As… buttons are used in order to save the configured Analyzing Model with all of its settings. All saved Analyzing Models are shown by the **ANS Explorer** (“Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Model Files” branch) and can be accessed (e.g. started, stopped, opened, …) from there.
- For a more detailed description of all available functionalities of the **AME Standard T001** and of the available Analyzing Functions, refer to the reference manual of the “Analyzing System”, chapter “AME Standard T001” and chapter “Analyzing Functions”.

Try it out

- ➔ Create a new Analyzing Model.
- ➔ Build an easy Analyzing Model which takes two online data from the **MDS Explorer**, adds them together and outputs the result.
- ➔ Move the functions around within the Analyzing Model.
- ➔ Change the visualization of some functions, e.g. turn on/off the values of the input arguments or expand and collapse the function.
- ➔ Save the Analyzing Model. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to “Hello Analyzing Model” and press OK in order to save it to the disk.

The new Analyzing Model now is displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Model Files” branch of the **ANS Explorer**.

➔ Take a look onto the Analyzing Models which are provided via the example location (see point 6.5) in order to see various application examples about the capabilities and usage of Analyzing Models.

Ready

At this point you know how to create a new Analyzing Model which can be used in order to perform calculations based on online and/or offline data.

6.13.3 Opening of existing Analyzing Models

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.

Functionality

- Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Model Files” branch.
- Drag one of the below Analyzing Models with the left mouse into the empty space at the right side of the **ANS Explorer**. An **Analyzing Model Editor** of proper type is opened automatically and displays the dropped Analyzing Model.

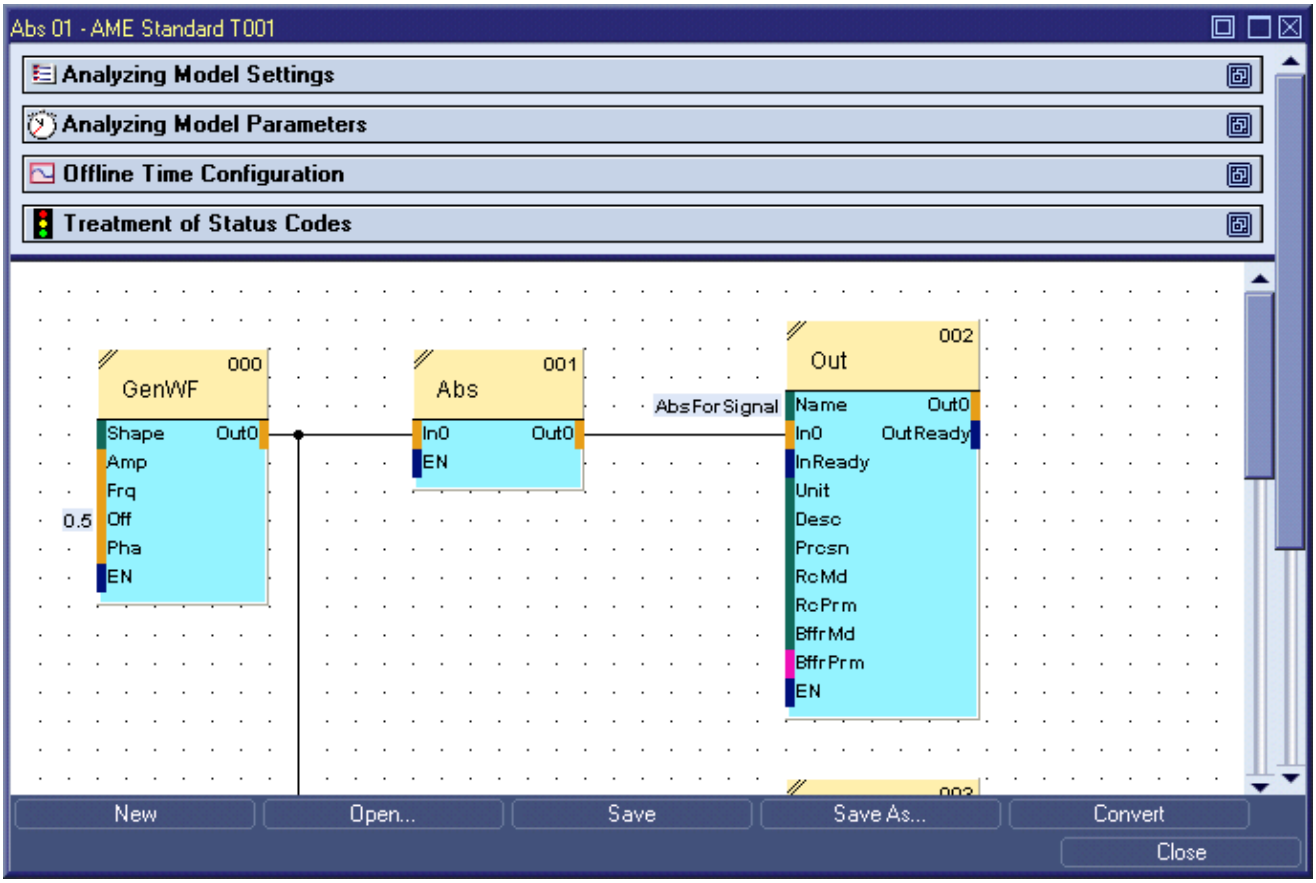


*Figure 36: **AME Standard T001** with an opened Analyzing Model*

- The opened Analyzing Model can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited Analyzing Model with all of its settings.

Try it out

➔ Open an existing Analyzing Model.

➔ Enter another storage location and change the Model Description within the Analyzing Model Settings table.

➔ Add an additional function to the Analyzing Model and output also this additional result of the calculation.

➔ Save the Analyzing Model via the Save button.

The modified Analyzing Model is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Model Files” branch of the **ANS Explorer**.

Ready

At this point you know how to open and modify an existing Analyzing Model.

6.13.4 Starting and Stopping of Analyzing Models

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” and “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branches of the **MDS Explorer**.

Functionality

- Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Model Files” branch.
- Click onto any of the present Analyzing Models with the right mouse button in order to open the context menu.
- In order to start the currently selected Analyzing Model, choose Start from the context menu.
  - Multiple Analyzing Models can be running simultaneously and without any dependencies between each other.
- The configured data of each started Analyzing Model is found within the **MDS Explorer** below its “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” (in case of output data class “Online Data”) or “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” (in case of output data class “Offline Data”) branch. From there, the data can be moved to all systems of ***X-Tools*** for further processing via Drag&Drop.
- In order to stop a currently running Analyzing Model, call the context menu of this Analyzing Model and choose Stop.
- There are several conditions in which an Analyzing Model can not run after it has been started, e.g. in case the input data is not available at the moment or in case the output data can not be created because it is in use already. When an Analyzing Model can not run after it has been started, it is being suspended and automatically resumed as soon as the error condition has disappeared.
- In case at least one Analyzing Model has been started by the user, the status bar of the ***X-Tools Client*** displays the current status of Analyzing Models via the according, colored icon. The tool tip of this icon provides detailed information about the number of currently started and suspended Analyzing Models.
  - Note that the status bar always displays the status of exactly one ***X-Tools Server***, thus also the icon for the current status of Analyzing Models displays the status of only the currently chosen ***X-Tools Server***.

Try it out

➔ Start an existing Analyzing Model. Ensure that the configured input data is available within the **MDS Explorer** and that the configured output data does not exist within the **MDS Explorer** yet.

The output data which has been configured within the started Analyzing Model becomes available within the **MDS Explorer**.

➔ Stop the currently running Analyzing Model.

The data which has been configured within the started Analyzing Model is being removed from the **MDS Explorer**.

Ready

At this point you know how to start and stop existing Analyzing Models in order to start and stop the calculations based on the currently available online and/or offline data.

6.14 Analyzing of Data via Analyzing Scripts

6.14.1 General Description

In this chapter you will learn how to analyze data from the **MDS Explorer** with the **Analyzing System** via Analyzing Scripts. An Analyzing Script contains all of the information which is needed in order to perform calculations based on the specified input data. This information includes the order of execution and the parameters of all to-be-called (Analyzing and User) functions. Multiple Analyzing Scripts can be started and running simultaneously at any time. In addition, the data which are defined as output data within a started Analyzing Script are available to all other modules of ***X-Tools*** via the **MDS Explorer**.

Examples of Analyzing Scripts are found within the example Configuration File location which is being delivered together with ***X-Tools***.

This chapter contains the following topics:

- **Creation of a new Analyzing Script**
- **Opening of existing Analyzing Scripts**
- **Starting and Stopping of Analyzing Scripts**

6.14.2 Creation of a new Analyzing Script

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Storage System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” and “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branches of the **MDS Explorer**.

Functionality

- Method 1 for the creation of a new Analyzing Script:
  - Within the **ANS Explorer**, expand the “Main -> Editors -> Analyzing Script Editors” branch.
  - Drag the **ASE Standard T001** with the left mouse into the empty space at the right side of the **ANS Explorer**. An empty **ASE Standard T001** is opened automatically.
  - Press the New button in order to create a new Analyzing Script. The new Analyzing Script is displayed by the **ASE Standard T001** with default values.
  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment. It can be used for the creation and configuration of Analyzing Scripts for cases where the target ***X-Tools Server*** is not present at the moment.
- Method 2 for the creation of a new Analyzing Script:
  - This method works only in case the **ASE Standard T001** is defined as “Default Editor for Functions” within the **Global Options Editor** of the ***X-Tools Client***.
  - Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> Analyzing Functions” branch.
  - Expand also the “Arithmetic” branch and drag a below Analyzing Function (e.g. the “Abs” Analyzing Function) into the empty **Analyzing System** workspace. An **ASE Standard T001** with a default Analyzing Script which also contains the dropped Analyzing Function is opened automatically.
  - Instead of an Analyzing Function from the “Main -> Servers -> *YourServer* -> Libraries -> Analyzing Functions” branch, also a User Function from the Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch can be dragged into the empty **Analyzing System**.
  - This method works only in case the target ***X-Tools Server*** is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new Analyzing Script.

- Method 3 for the creation of a new Analyzing Script:
  - This method works only in case the **ASE Standard T001** is defined as “Default Editor for Functions” within the **Global Options Editor**.
  - Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer* -> Online Data” or the “Main -> Servers -> *YourServer* -> Offline Data” branch.
  - Drag an online or offline data from the **MDS Explorer** into the empty **Analyzing System** workspace. An **ASE Standard T001** with a default Analyzing Script which also contains the dropped data is opened automatically.
  - This method works only in case the target data is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new Analyzing Script.
- The following screenshot shows an example of a new Analyzing Script.

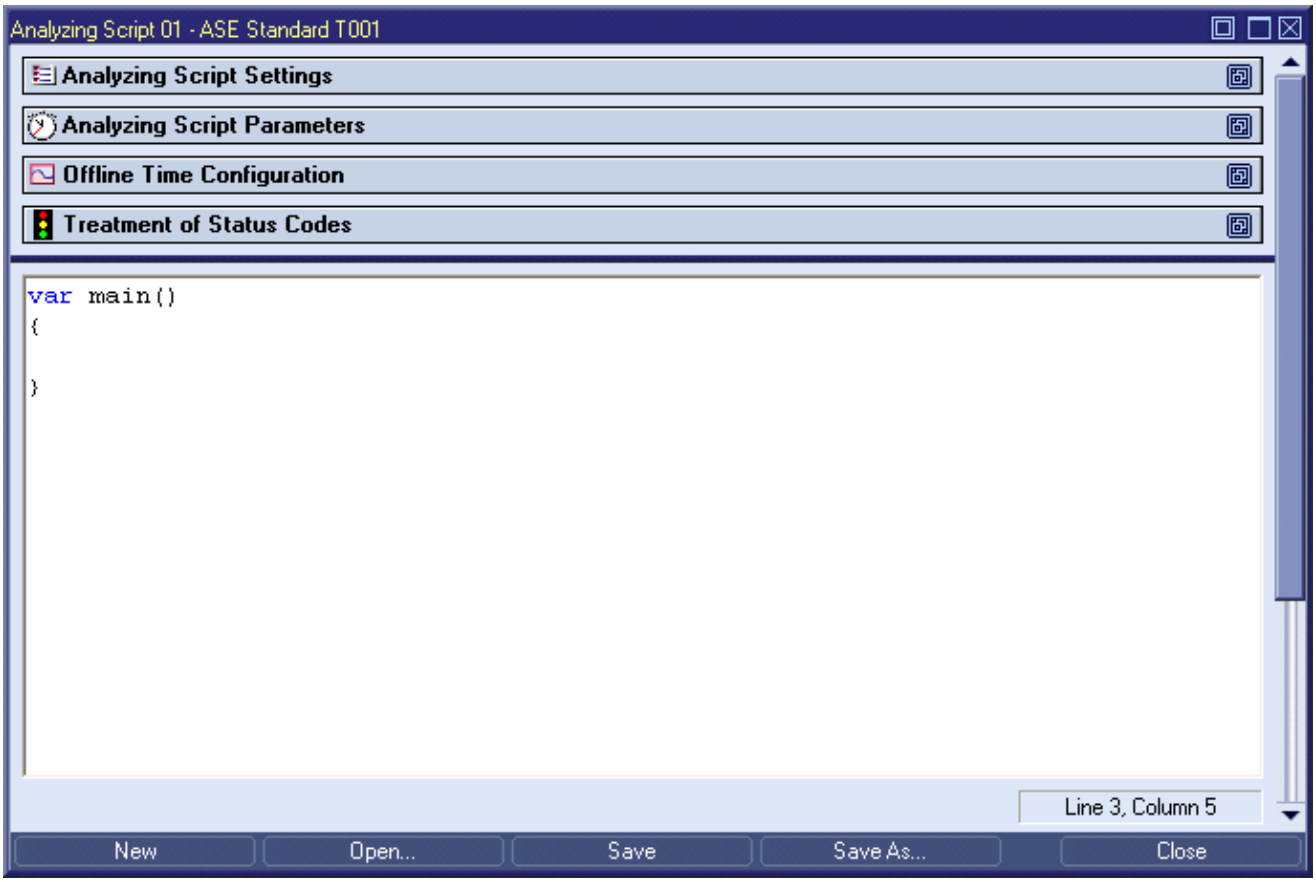


*Figure 37: **ASE Standard T001** with a new Analyzing Script*

- The Analyzing Script Settings table at the top of the **ASE Standard T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- Within the Analyzing Script Parameters table, the basic parameters for the execution of the Analyzing Script are being defined.
  - The default configuration of Analyzing Scripts typically allows executing the defined analysis without additional changes to the parameters.
  - For a detailed description of the meaning and usage of the parameters of Analyzing Scripts, refer to the reference manual of the “Analyzing System”, chapter “Analyzing Script Editors”.

- Within the Offline Time Configuration table, the time interval onto which the calculation shall be applied is being defined. The contents of this table can be configured only in case the output data class is specified as offline data.
  - The default configuration of Analyzing Scripts typically allows executing the defined analysis without additional changes to the offline time configuration.
  - For a detailed description of the meaning and usage of the offline time configuration of Analyzing Scripts, refer to the reference manual of the “Analyzing System”, chapter “Analyzing Script Editors”.
- Within the Treatment of Status Codes table, the handling of occurring status codes is being defined.
  - The default configuration of Analyzing Scripts typically allows executing the defined analysis without additional changes to the treatment of status codes.
  - For a detailed description of the meaning and usage of the treatment of status codes of Analyzing Scripts, refer to the reference manual of the “Analyzing System”, chapter “Analyzing Script Editors”.
- In order to add input data to the Analyzing Script, simply drag the desired data from the **MDS Explorer** into the Analyzing Script.
- In order to add Analyzing Functions to the Analyzing Script, simply drag the desired Analyzing Function from the **ANS Explorer** into the Analyzing Script.
- In order to add User Functions to the Analyzing Script, simply drag the desired User Function from the **ANS Explorer** into the Analyzing Script.
- Functions are being called in the order in which they are found in the Analyzing Script. Via variables and/or via nested calls the output values of each function can be provided as input values to other functions.
- An Analyzing Script can work either only on online data, only on offline data or on online and offline data simultaneously. The class of the resulting data (online or offline) can be configured via the Analyzing Script Parameters table.
- Analyzing Functions from the category “Input” are being used in order to get (online or offline) data from the system into the Analyzing Script. Analyzing Functions from the category “Output” are being used in order to put (online or offline) data from the Analyzing Script back to the system.
- The Save and Save As… buttons are used in order to save the configured Analyzing Script with all of its settings. All saved Analyzing Scripts are shown by the **ANS Explorer** (“Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Script Files” branch) and can be accessed (e.g. started, stopped, opened, …) from there.
- In addition to the time slice based operating mode, Analyzing Scripts also can be executed in sequential processing. The sequential processing allows further operations like manipulating of time intervals and conditional statements within the Analyzing Script. For a more detailed description of all available functionalities of the **ASE Standard T001** and of the available Analyzing Functions, refer to the reference manual of the “Analyzing System”, chapter “ASE Standard T001” and chapter “Analyzing Functions”.

Try it out

➔ Create a new Analyzing Script.

➔ Build an easy Analyzing Script which takes two online data from the **MDS Explorer**, adds them together and outputs the result.

➔ Save the Analyzing Script. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to “Hello Analyzing Script” and press OK in order to save it to the disk.

The new Analyzing Script now is displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Script Files” branch of the **ANS Explorer**.

➔ Take a look onto the Analyzing Scripts which are provided via the example location (see point 6.5) in order to see various application examples about the capabilities and usage of Analyzing Scripts.

Ready

At this point you know how to create a new Analyzing Script which can be used in order to perform calculations based on online and/or offline data.

6.14.3 Opening of existing Analyzing Scripts

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.

Functionality

- Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Script Files” branch.
- Drag one of the below Analyzing Scripts with the left mouse into the empty space at the right side of the **ANS Explorer**. An **Analyzing Script Editor** of proper type is opened automatically and displays the dropped Analyzing Script.

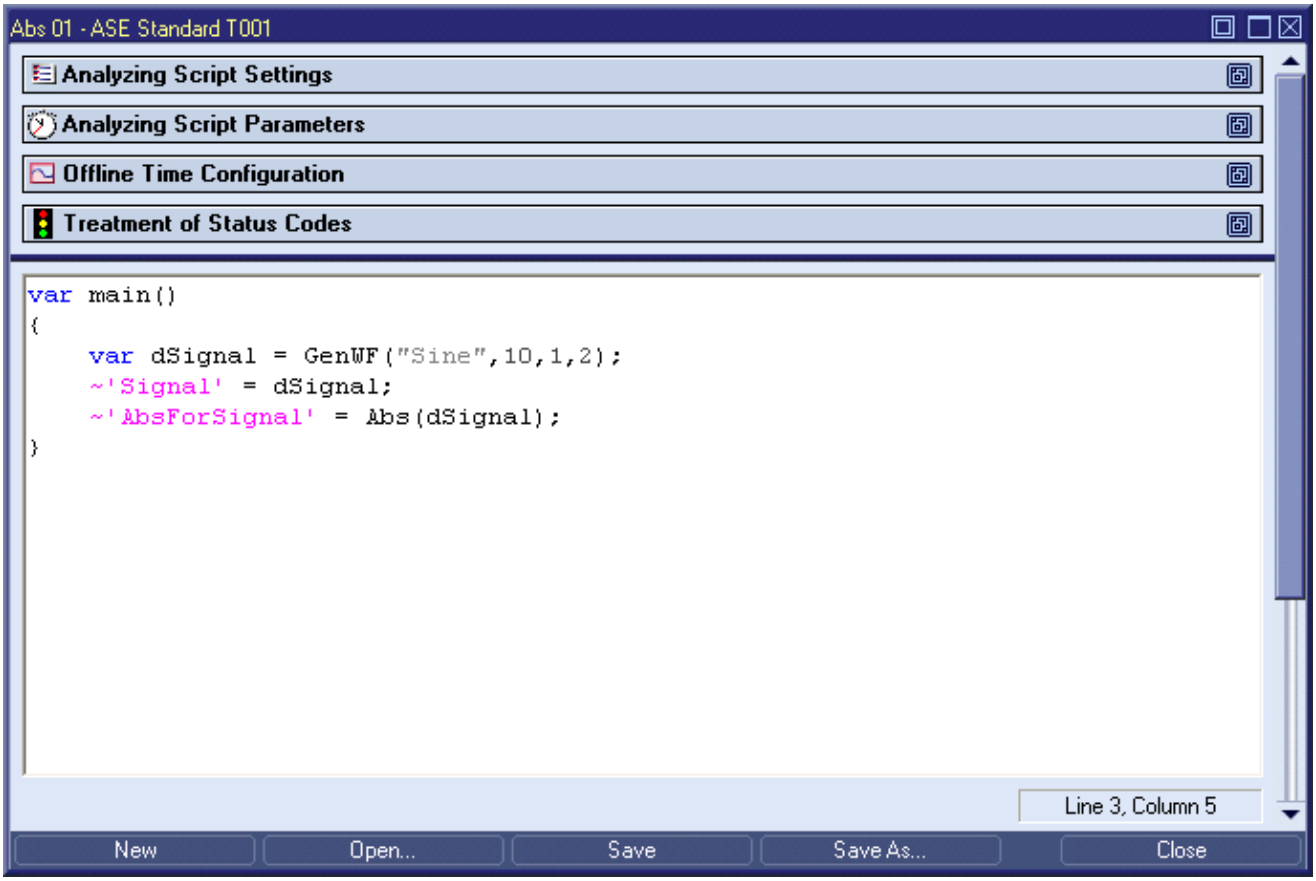


*Figure 38: **ASE Standard T001** with an opened Analyzing Script*

- The opened Analyzing Script can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited Analyzing Script with all of its settings.

Try it out

➔ Open an existing Analyzing Script.

➔ Enter another storage location and change the Script Description within the Analyzing Script Settings table.

➔ Add an additional function to the Analyzing Script and output also this additional result of the calculation.

➔ Save the Analyzing Script via the Save button.

The modified Analyzing Script is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Script Files” branch of the **ANS Explorer**.

Ready

At this point you know how to open and modify an existing Analyzing Script.

6.14.4 Starting and Stopping of Analyzing Scripts

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” and “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branches of the **MDS Explorer**.

Functionality

- Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Script Files” branch.
- Click onto any of the present Analyzing Scripts with the right mouse button in order to open the context menu.
- In order to start the currently selected Analyzing Script, choose Start from the context menu.
  - Multiple Analyzing Scripts can be running simultaneously and without any dependencies between each other.
- The configured data of each started Analyzing Script is found within the **MDS Explorer** below its “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” (in case of output data class “Online Data”) or “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” (in case of output data class “Offline Data”) branch. From there, the data can be moved to all systems of ***X-Tools*** for further processing via Drag&Drop.
- In order to stop a currently running Analyzing Script, call the context menu of this Analyzing Script and choose Stop.
- There are several conditions in which an Analyzing Script can not run after it has been started, e.g. in case the input data is not available at the moment or in case the output data can not be created because it is in use already. When an Analyzing Script can not run after it has been started, it is being suspended and automatically resumed as soon as the error condition has disappeared.
- In case at least one Analyzing Script has been started by the user, the status bar of the ***X-Tools Client*** displays the current status of Analyzing Scripts via the according, colored icon. The tool tip of this icon provides detailed information about the number of currently started and suspended Analyzing Scripts.
  - Note that the status bar always displays the status of exactly one ***X-Tools Server***, thus also the icon for the current status of Analyzing Scripts displays the status of only the currently chosen ***X-Tools Server***.

Try it out

➔ Start an existing Analyzing Script. Ensure that the configured input data is available within the **MDS Explorer** and that the configured output data does not exist within the **MDS Explorer** yet.

The output data which has been configured within the started Analyzing Script becomes available within the **MDS Explorer**.

➔ Stop the currently running Analyzing Script.

The data which has been configured within the started Analyzing Script is being removed from the **MDS Explorer**.

Ready

At this point you know how to start and stop existing Analyzing Scripts in order to start and stop the calculations based on the currently available online and/or offline data.

6.15 User Functions for the Encapsulation of Analysis Tasks

6.15.1 General Description

In this chapter you will learn how to create User Functions. User Functions can be used in order to combine multiple (Analyzing and User) functions to a new entity. Uses cases for User Functions are:

- The same or similar analysis tasks are being used by different Analyzing Models or Analyzing Scripts.
- Complex analysis tasks can be represented by a single User Function, which improves the readability of Analyzing Models and Analyzing Scripts.
- User Functions can improve the maintainability. All Analyzing Models and Analyzing Scripts which contain a User Function use the latest version of this User Function whenever the Analyzing Model or Analyzing Script is being used. Thus, changing of one User Function can be used in order to update all of the Analyzing Models and Analyzing Scripts which use this User Function.
- The internal calculations shall be hidden from other users. User Functions can be protected agains reading in order to secure the contained know-how.

Like Analyzing Functions, User Functions can not be executed stand-alone. They perform their configured tasks only during the processing of their parent Analyzing Model or Analyzing Script. In addition to the full name and to the short name, also a category can be assigned to each User Function. The category is used by the **ANS Explorer** in order to group together User Functions of the same category.

Within the **UFE Standard T001**, a User Function contains (Analyzing and User) functions and links like an Analyzing Model. Outside the **UFE Standard T001**, a User Function can be used by Analyzing Models and Analyzing Scripts like any of the standard Analyzing Functions. Multiple User Functions can be maintained and/or executed within Analyzing Models and Analyzing Scripts simultaneously.

Examples of User Functions are found within the example Configuration File location which is being delivered together with ***X-Tools***.

This chapter contains the following topics:

- **Creation of a new User Function**
- **Opening of existing User Functions**
- **Using of User Functions within Analyzing Models**
- **Using of User Functions within Analyzing Scripts**
- **Behavior of User Functions**
- **Examples of User Functions**

6.15.2 Creation of a new User Function

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” and “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branches of the **MDS Explorer**.

Functionality

- Method 1 for the creation of a new User Function:
  - Within the **ANS Explorer**, expand the “Main -> Editors -> User Function Editors” branch.
  - Drag the **UFE Standard T001** with the left mouse into the empty space at the right side of the **ANS Explorer**. An empty **UFE Standard T001** is opened automatically.
  - Press the New button in order to create a new User Function. The new User Function is displayed by the **UFE Standard T001** with default values.

  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment. It can be used for the creation and configuration of User Functions for cases where the target ***X-Tools Server*** is not present at the moment.

- Method 2 for the creation of a new User Function:
  - This method works only in case the **UFE Standard T001** is defined as “Default Editor for Functions” within the **Global Options Editor** of the ***X-Tools Client***.
  - Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> Analyzing Functions” branch.
  - Expand also the “Arithmetic” branch and drag a below Analyzing Function (e.g. the “Abs” Analyzing Function) into the empty **Analyzing System** workspace. A **UFE Standard T001** with a default User Function which also contains the dropped Analyzing Function is opened automatically.
  - Instead of an Analyzing Function from the “Main -> Servers -> *YourServer* -> Libraries -> Analyzing Functions” branch, also a User Function from the Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch can be dragged into the empty **Analyzing System**.
  - This method works only in case the target ***X-Tools Server*** is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new User Function.
- Method 3 for the creation of a new User Function:
  - This method works only in case the **UFE Standard T001** is defined as “Default Editor for Functions” within the **Global Options Editor** of the ***X-Tools Client***.
  - Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer* -> Online Data” or the “Main -> Servers -> *YourServer* -> Offline Data” branch.
  - Drag an online or offline data from the **MDS Explorer** into the empty **Analyzing System** workspace. A **UFE Standard T001** with a default User Function which also contains the dropped data is opened automatically.
  - This method works only in case the target data is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new User Function.
- Method 4 for the creation of a new User Function:
  - Open an existing Analyzing Model or create a new Analyzing Model.
  - Within the **AME Standard T001**, press the Convert button.
  - The Analyzing Model is converted into an identical User Function automatically. The new User Function is opened within a **UFE Standard T001**. In order to become able to use the new User Function, valid information must be provided for the full name and short name of the User Function and the input and output interfaces must be defined.

- The following screenshot shows an example of a new User Function.

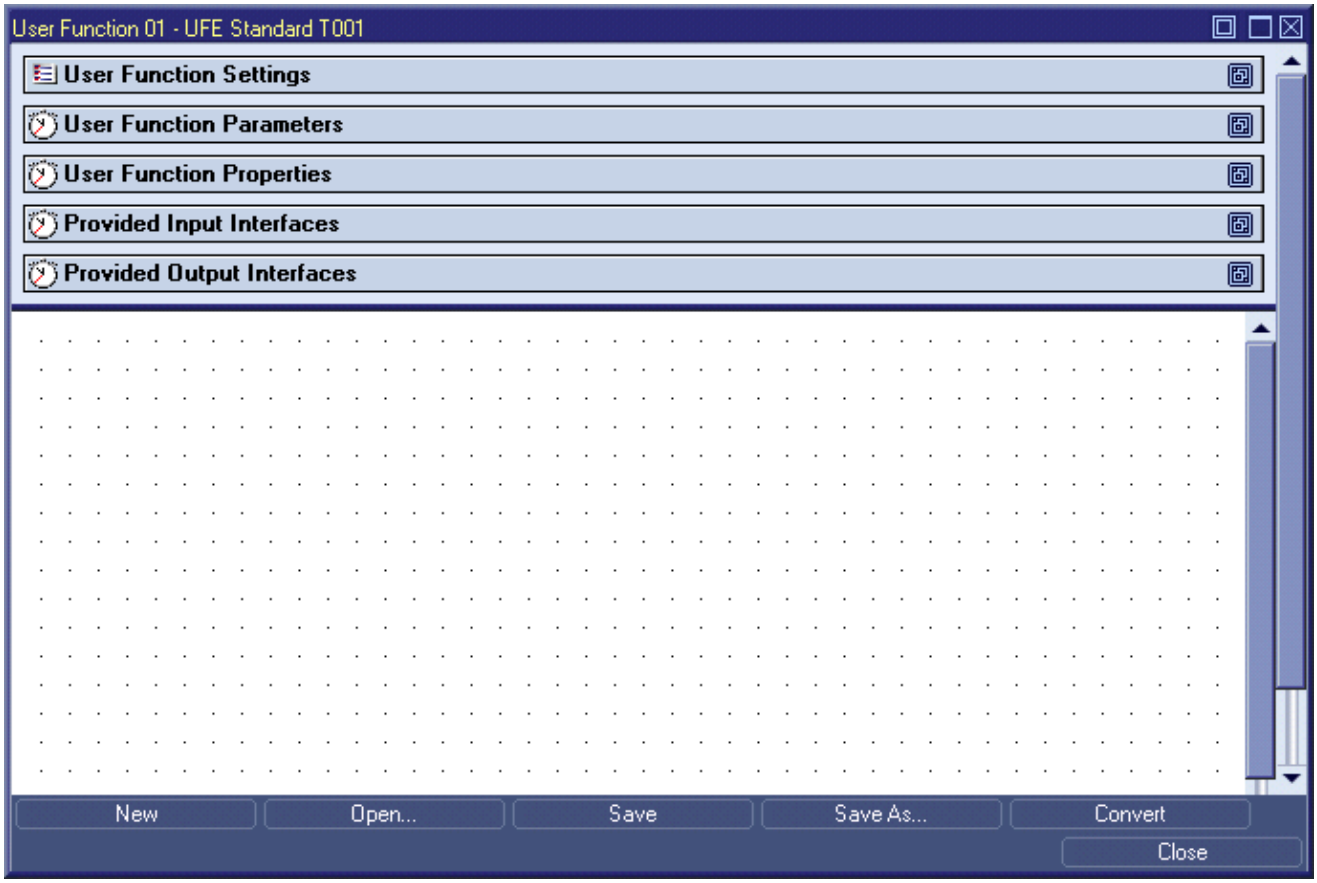


*Figure 39: **UFE Standard T001** with a new User Function*

- The User Function Settings table at the top of the **UFE Standard T001** displays the general file information and allows entering additional file attributes like a description and the name of the author of the file and his company.
- The User Function Parameters table allows specifying a password for the User Function. In case a User Function is protected by a password, it can not be opened within the **UFE Standard T001** without specifying of the correct password. In addition to the password it also can be specified whether the User Function shall forward the log entries of its contained functions when it is being used by an Analyzing Model or Analyzing Script.
- The User Function Properties table displays basic information about the User Function like the operating modes which will it support. In addition, it allows entering the names and the description as well as the parent category of the User Function.
- Within the Provided Input Interfaces table, all of the input interfaces which are defined for the User Function are being displayed and can be edited.
- Within the Provided Output Interfaces table, all of the output interfaces which are defined for the User Function are being displayed and can be edited.
- In order to add input data to the User Function, simply drag the desired data from the **MDS Explorer** into the User Function.
- In order to add Analyzing Functions to the User Function, simply drag the desired Analyzing Function from the **ANS Explorer** into the User Function.
- In order to add User Functions to the User Function, simply drag the desired User Function from the **ANS Explorer** into the User Function.
- (Analyzing and User) functions can be connected together via links. Each output of a function can be connected to one or more inputs of functions by clicking onto the output argument first and by clicking onto the input argument next. Each created link transports the output value from its source function to the input argument of its destination function.
- All of the currently present functions can be arranged within the User Function via Drag&Drop.

- A double-click onto a function opens the **Analyzing Function Properties** dialog which can be used in order to specify the values of the input arguments of the function.
- Via the context menu, functions can be cut/copied/pasted/removed and their visualization can be modified.
- Analyzing Functions from the category “Input” are being used in order to get (online or offline) data from the system into the User Function. Analyzing Functions from the category “Output” are being used in order to put (online or offline) data from the User Function back to the system.
- Each User Function must provide inputs and outputs which can be used later by the Analyzing Model or Analyzing Script which embeds the User Function. In order to add an input which is visible from outside, call the context menu above the input argument which shall be visible outside and choose “Add Input Interface”. In order to add an output which is visible from outside, call the context menu above the output argument and choose “Add Output Interface”.
- After an input or output interface has been created, the name, unit, description and default value can be edited within the Provided Input Interfaces and Provided Output Interfaces table.
- A User Function is not limited to a certain operating mode, calculation mode, cycle time or output data class. All of these parameters are configured only within the parent Analyzing Model or Analyzing Script in the later step. Depending to the contained functions, a User Function may not be able to operate in all operating modes and calculation modes or it may not be able to work together with all output data classes.
- Before a User Function can be saved, the Full Name and Short Name of the User Function must be specified within the User Function Properties table.
- The Save and Save As… buttons are used in order to save the configured User Function with all of its settings. All saved User Functions are shown by the **ANS Explorer** (“Main -> Servers -> *YourServer* -> *YourLocation* -> User Function Files” branch) and can be accessed from there. In addition, each User Function also creates one entry below the “Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch of the **ANS Explorer**.
- For a more detailed description of all available functionalities of the **UFE Standard T001** and of the available Analyzing Functions, refer to the reference manual of the “Analyzing System”, chapter “UFE Standard T001” and chapter “Analyzing Functions”.

Try it out

➔ Create a new User Function.

➔ Build an easy User Function which adds two data together and subtracts a value of 10 from the result.

➔ This User Function needs only two Analyzing Functions, Add() and Sub(). In order to provide the external interface, the two inputs of Add() are configured as input interfaces (call the context menu for each input and choose Add Input Interface) and the output of Sub() is configured as output interface (call the context menu for the output and choose Add Output Interface).

➔ Move the functions around within the User Function.

➔ Change the visualization of some functions, e.g. turn on/off the values of the input arguments or expand and collapse the function.

➔ Within the User Function Properties table, set a Full Name of “HelloWorld”, a Short Name of “HelloW” and a Parent Category of “Tutorial”.

➔ The User Function looks like in the following example:

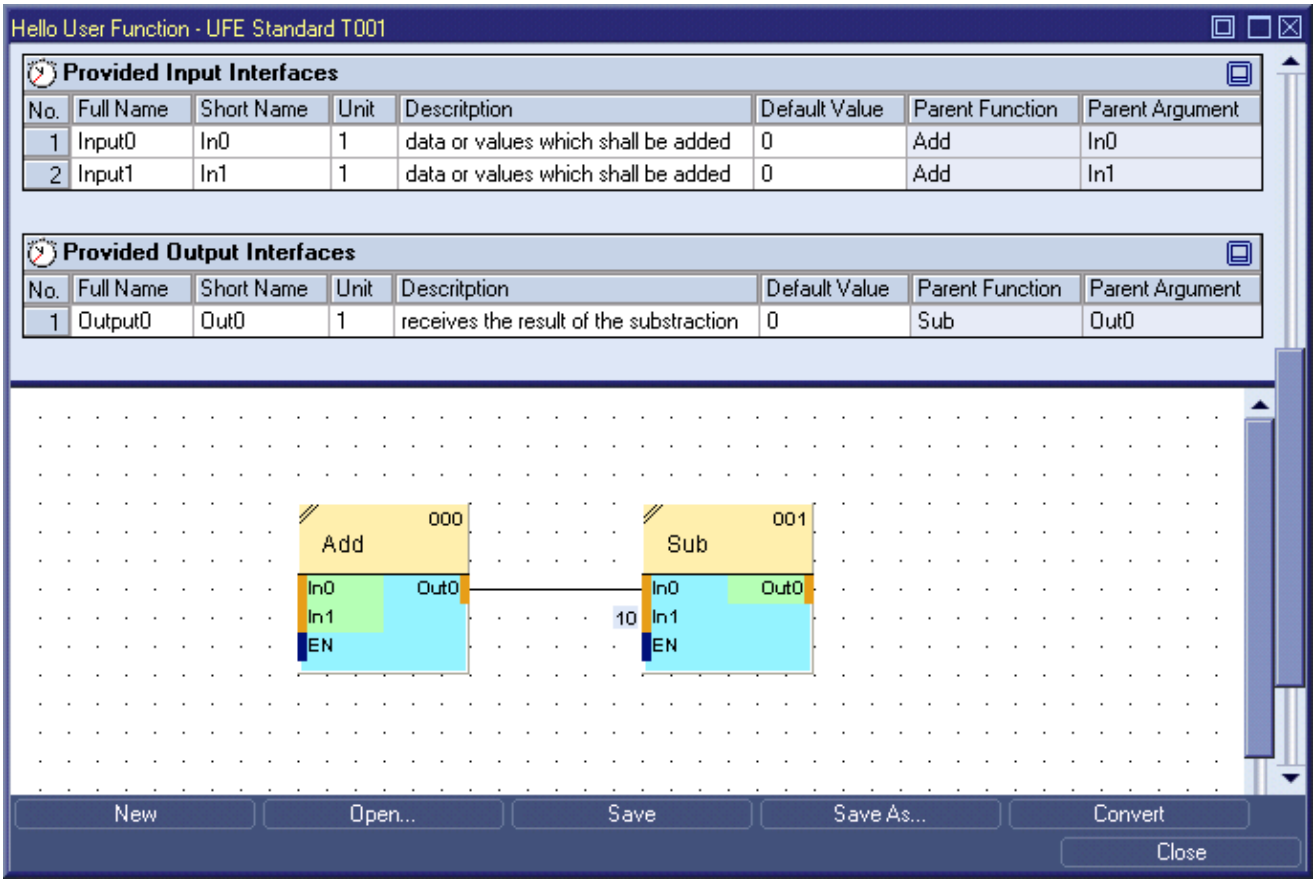


*Figure 40: **UFE Standard T001** with the HelloW() Example*

➔ Save the User Function. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to “Hello User Function” and press OK in order to save it to the disk.

The new User Function now is displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> User Function Files” branch of the **ANS Explorer**.

➔ Create another User Function. This time, add the two Analyzing Functions for the input and the one Analyzing Function for the output data directly into the User Function.

➔ In order to provide the external interface, specify input interfaces for all three “Name” input arguments of the In() and Out() Analyzing Functions.

➔ Within the Provided Input Interfaces table, assign short names like “In0” and “In1” for the “Name” input arguments of the In() Analyzing Functions and a short name like “Out0” for the “Name” input argument of the Out() Analyzing Function.

➔ Specify an output interface for the “Out0” output argument of the Out() Analyzing Function.

➔ Within the Provided Output Interfaces table, assign a short name like “Out1” for the “Out0” output argument of the Out() Analyzing Function.

➔ Within the User Function Properties table, set a Full Name of “HelloWorld2”, a Short Name of “HelloW2” and a Parent Category of “Tutorial”.

➔ The User Function looks like in the following example:

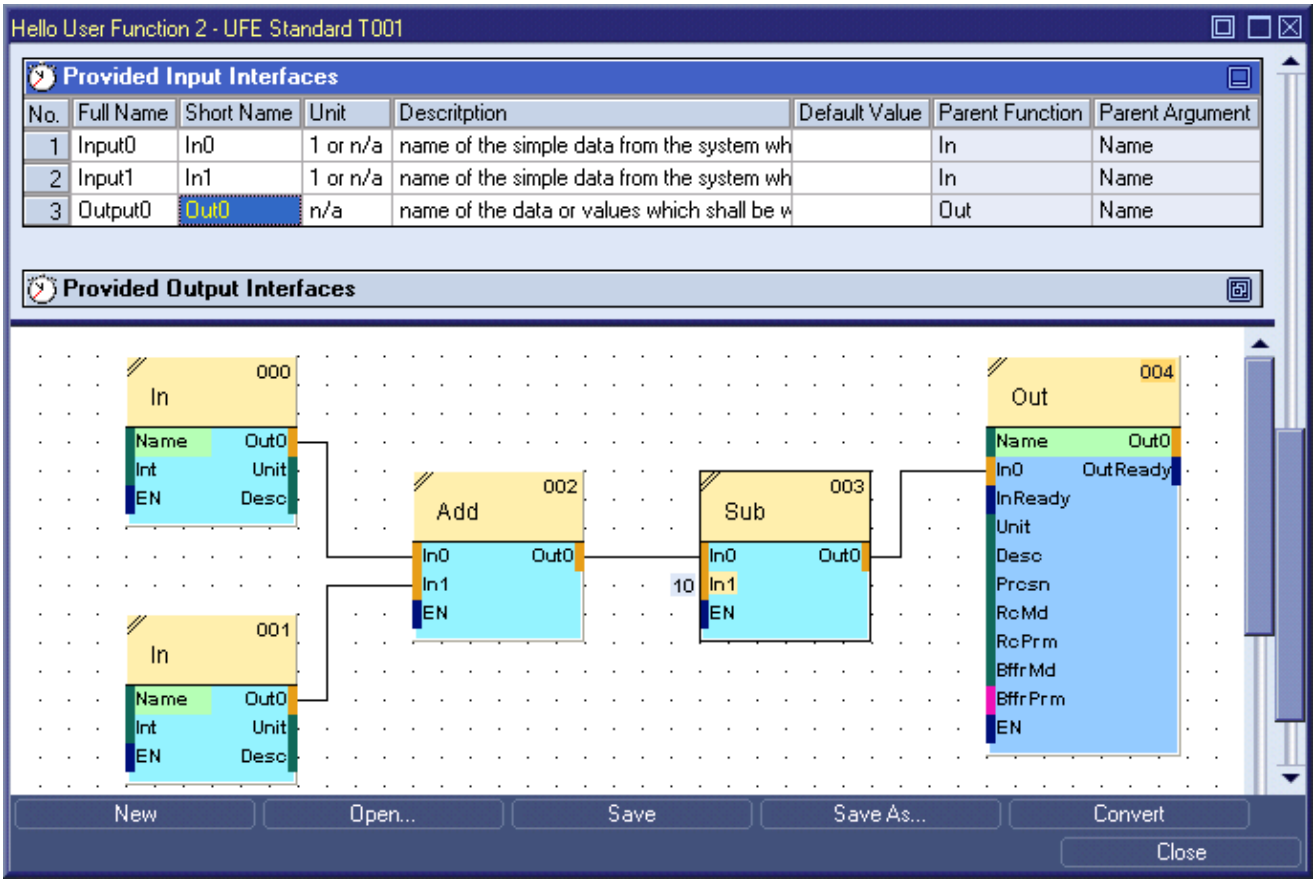

Hello User Function 2 - UFE Standard T001
Provided Input Interfaces
No. Full Name Short Name Unit Descritption Default Value Parent Function Parent Argument
1 Input0 In0 1 or n/a name of the simple data from the system wh In Name
2 Input1 In1 1 or n/a name of the simple data from the system wh In Name
3 Output0 Out0 n/a name of the data or values which shall be w Out Name
Provided Output Interfaces
000 In Name Out0 Int Unit EN Desc
001 In Name Out0 Int Unit EN Desc
002 Add In0 Out0 In1 EN
003 Sub In0 Out0 10 In1 EN
004 Out Name Out0 In0 OutReady InReady Unit Desc Prcsn RcMd RcPrm BffrMd BffrPrm EN
New
Open...
Save
Save As...
Convert
Close

*Figure 41: **UFE Standard T001** with the HelloW2() Example*

➔ Compared to the User Function “HelloW”, “HelloW2” provides less possibilities to the parent Analyzing Model or Analyzing Script to influence the processing of the input and output data, because “HelloW2” internally contains the In() and Out() Analyzing Functions which decide about the to-be-used interpolation and record modes. Depending to the target application, both methods for the embedding of input and output data can have their advantages and disadvantages and it must be decided for each application, which approach is more suitable.

Ready

At this point you know how to create a new User Function which can be used by Analyzing Models and Analyzing Scripts in order to perform calculations based on online and/or offline data.

6.15.3 Opening of existing User Functions

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.

Functionality

- Method 1 for the opening of an existing User Function:
  - Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> User Function Files” branch.
  - Drag one of the below User Functions with the left mouse into the empty space at the right side of the **ANS Explorer**. A **User Function Editor** of proper type is opened automatically and displays the dropped User Function.

- Method 2 for the opening of an existing User Function:
  - Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch.
  - Call the context menu for one of the below User Functions and choose Edit. A **User Function Editor** of proper type is opened automatically and displays the selected User Function.

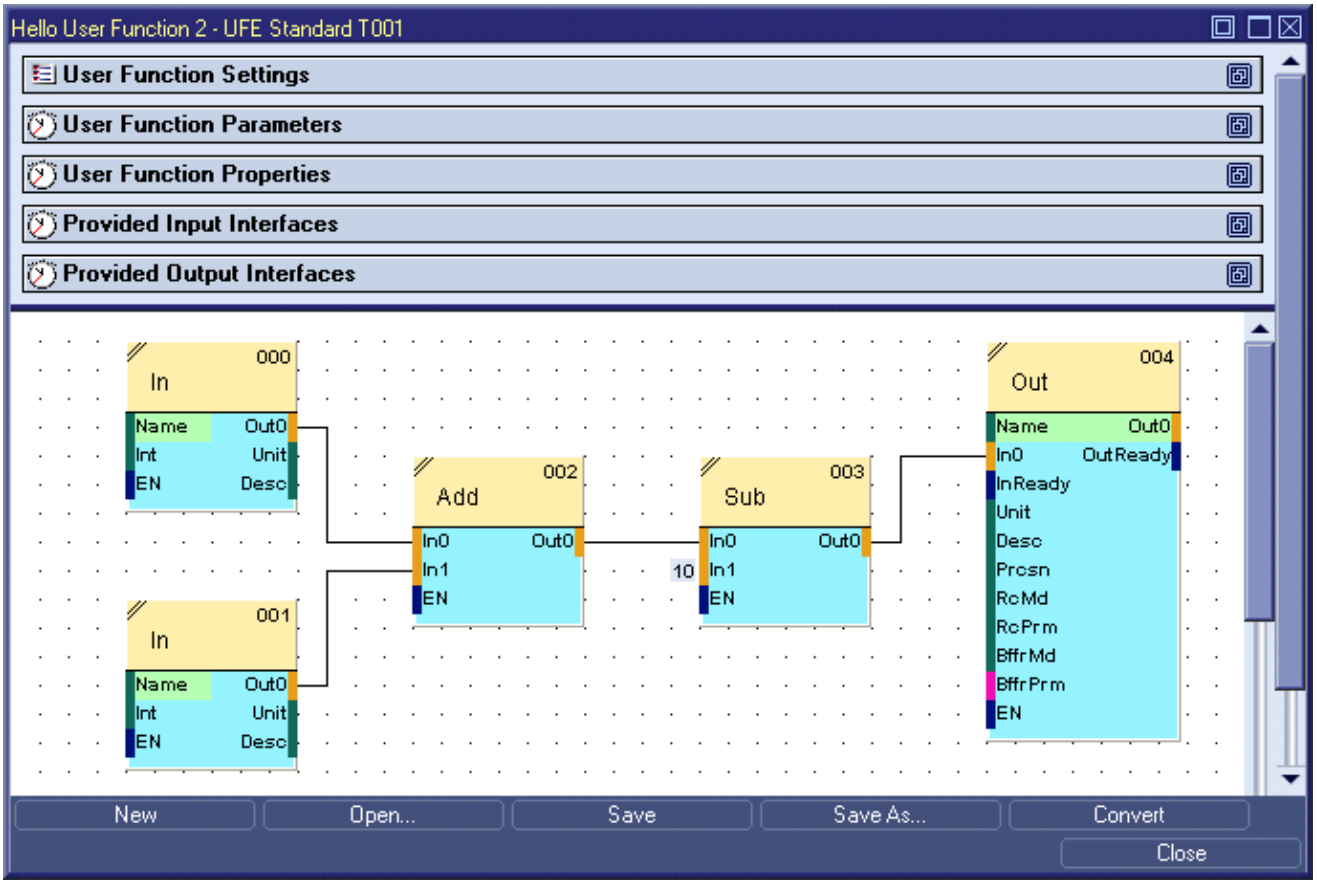


*Figure 42: **UFE Standard T001** with an opened User Function*

- The opened User Function can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited User Function with all of its settings.

Try it out

➔ Open an existing User Function.

➔ Enter another storage location and change the Model Description within the User Function Settings table.

➔ Add an additional function to the User Function and output also this additional result of the calculation.

➔ Save the User Function via the Save button.

The modified User Function is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> User Function Files” branch of the **ANS Explorer**.

Ready

At this point you know how to open and modify an existing User Function.

6.15.4 Using of User Functions within Analyzing Models

Provide some online Data

- Ensure that there is some online data available within the **MDS Explorer**.

Functionality

- Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch.
- The present User Functions can be dropped into the **AME Standard T001**.
- The input and output arguments of the User Function can be connected to any other (Analyzing and User) functions within the Analyzing Model.
- After the Analyzing Model has been completed, it can be saved and started.

Try it out

➔ Create a new, empty Analyzing Model.

➔ Drag&Drop two online data from the **MDS Explorer** into the **AME Standard T001**.

➔ Drag&Drop the User Function “HelloW”, which has been created during the previous tutorial, from the **ANS Explorer** into the **AME Standard T001**.

➔ Drag&Drop the Analyzing Function Out() from the **ANS Explorer** into the **AME Standard T001**.

➔ Connect the outputs of the two inputs to the inputs of HelloW() and the output of HelloW() to the input of Out(). Specify a name for the output data and save the Analyzing Model.

➔ The Analyzing Model looks like in the following example:

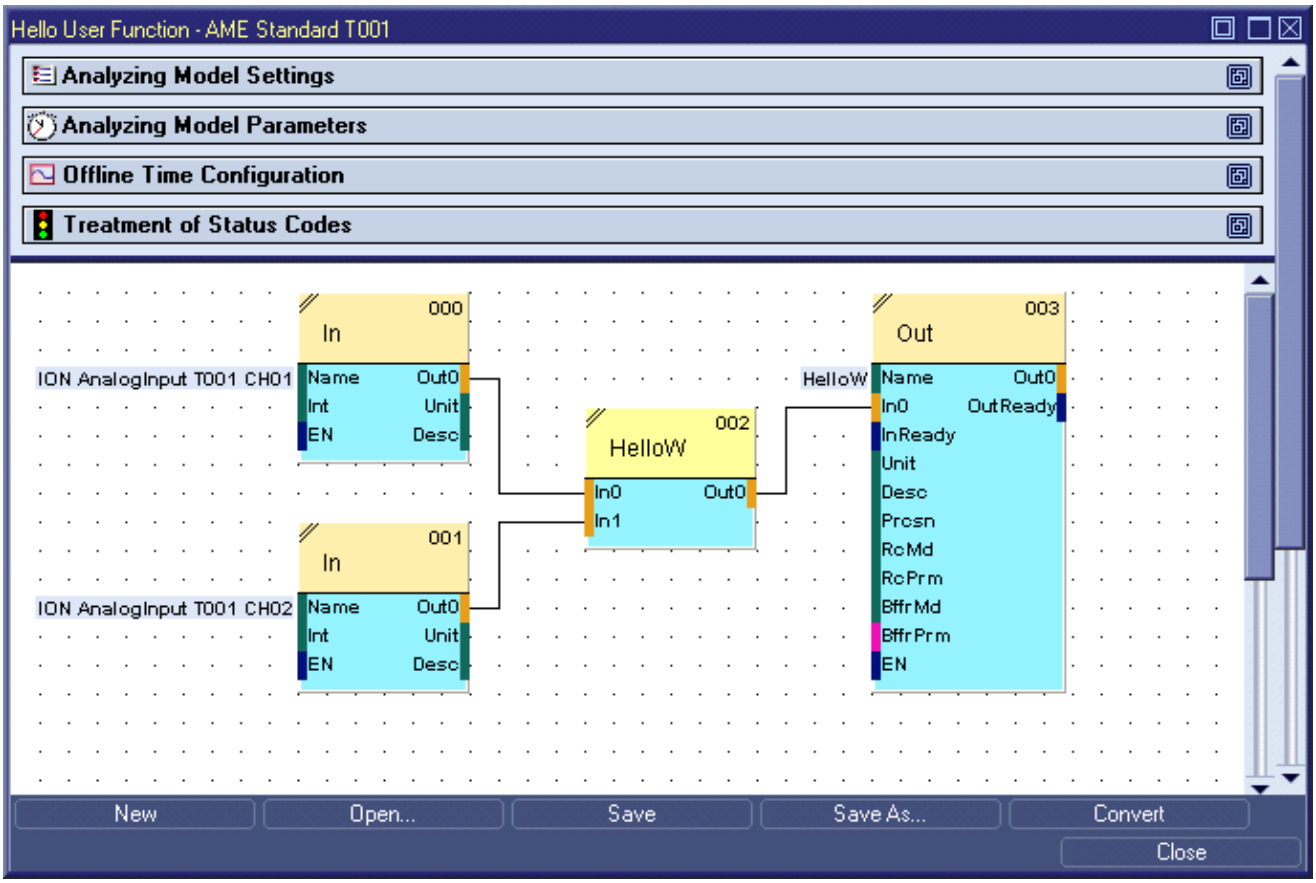


*Figure 43: **AME Standard T001** with the HelloW() User Function*

➔ Start the Analyzing Model.

The output data which has been configured within the started Analyzing Model becomes available within the **MDS Explorer**.

➔ Create another new, empty Analyzing Model.

➔ Drag&Drop the User Function “HelloW2”, which has been created during the previous tutorial, from the **ANS Explorer** into the **AME Standard T001**.

➔ Enter the name of an online data from the **MDS Explorer** for the “In0” argument of HelloW2() and the name of another online data for the “In1” argument.

➔ Enter a name for the “Out0” argument of HelloW2().

➔ The Analyzing Model looks like in the following example:

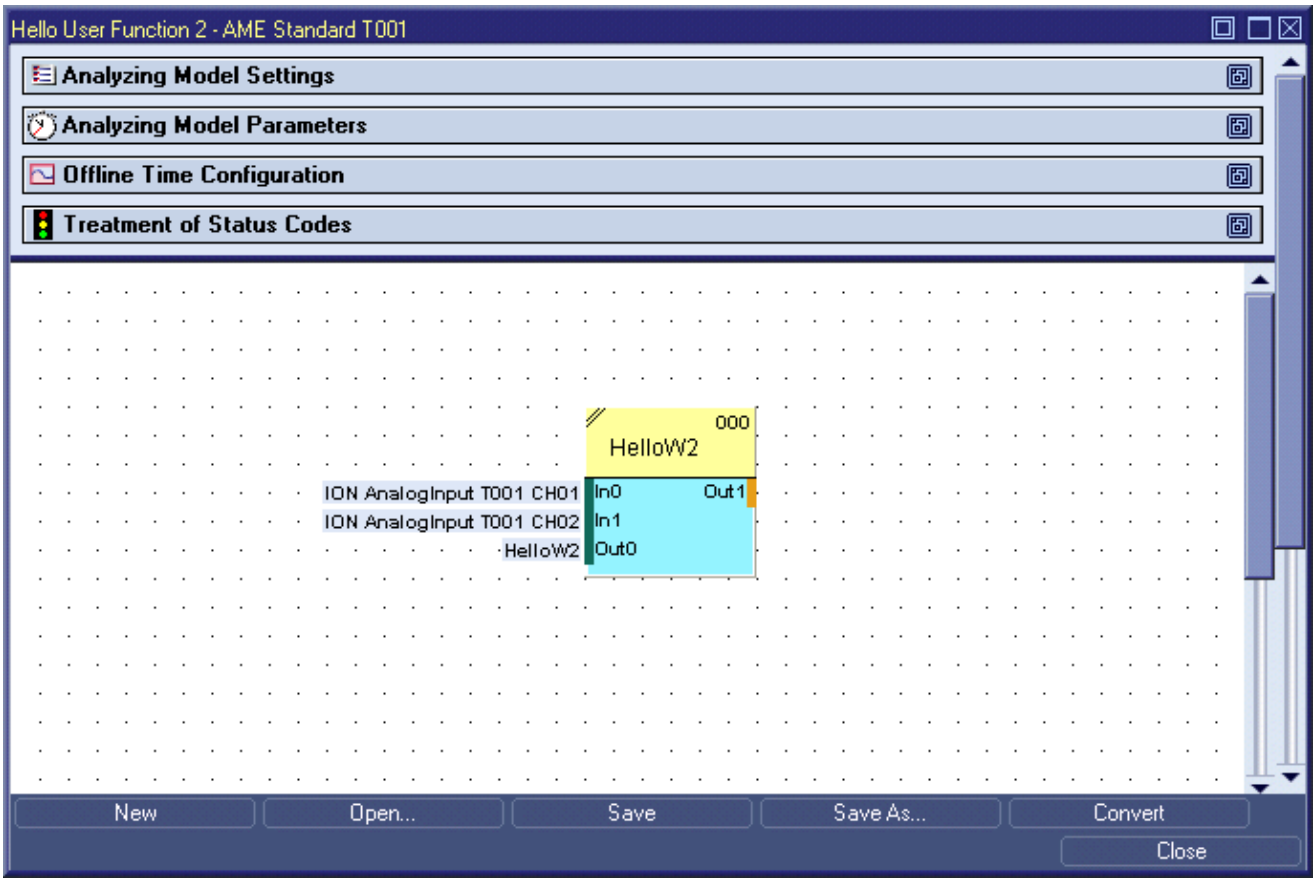


*Figure 44: **AME Standard T001** with the HelloW2() User Function*

➔ Start the Analyzing Model.

The output data which has been configured within the started Analyzing Model becomes available within the **MDS Explorer**.

➔ Compare the output of both Analyzing Models within the Monitoring System.

Both Analyzing Models produce identical output data.

➔ Stop both Analyzing Models.

The data which have been configured within the started Analyzing Models are being removed from the **MDS Explorer**.

Ready

At this point you know how to use a User Function within an Analyzing Model.

6.15.5 Using of User Functions within Analyzing Scripts

Provide some online Data

- Ensure that there is some online data available within the **MDS Explorer**.

Functionality

- Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch.
- The present User Functions can be dropped into the **ASE Standard T001**.
- The syntax for using of User Functions is identical to the syntax of Analyzing Functions.
- After the Analyzing Script has been completed, it can be saved and started.

Try it out

➔ Create a new, empty Analyzing Script.

➔ Drag&Drop the User Function “HelloW”, which has been created during the previous tutorial, from the **ANS Explorer** into the **ASE Standard T001**.

➔ Provide two online data from the **MDS Explorer** as parameters for HelloW().

➔ Specify a data name for the return value of HelloW().

➔ The Analyzing Script looks like in the following example:

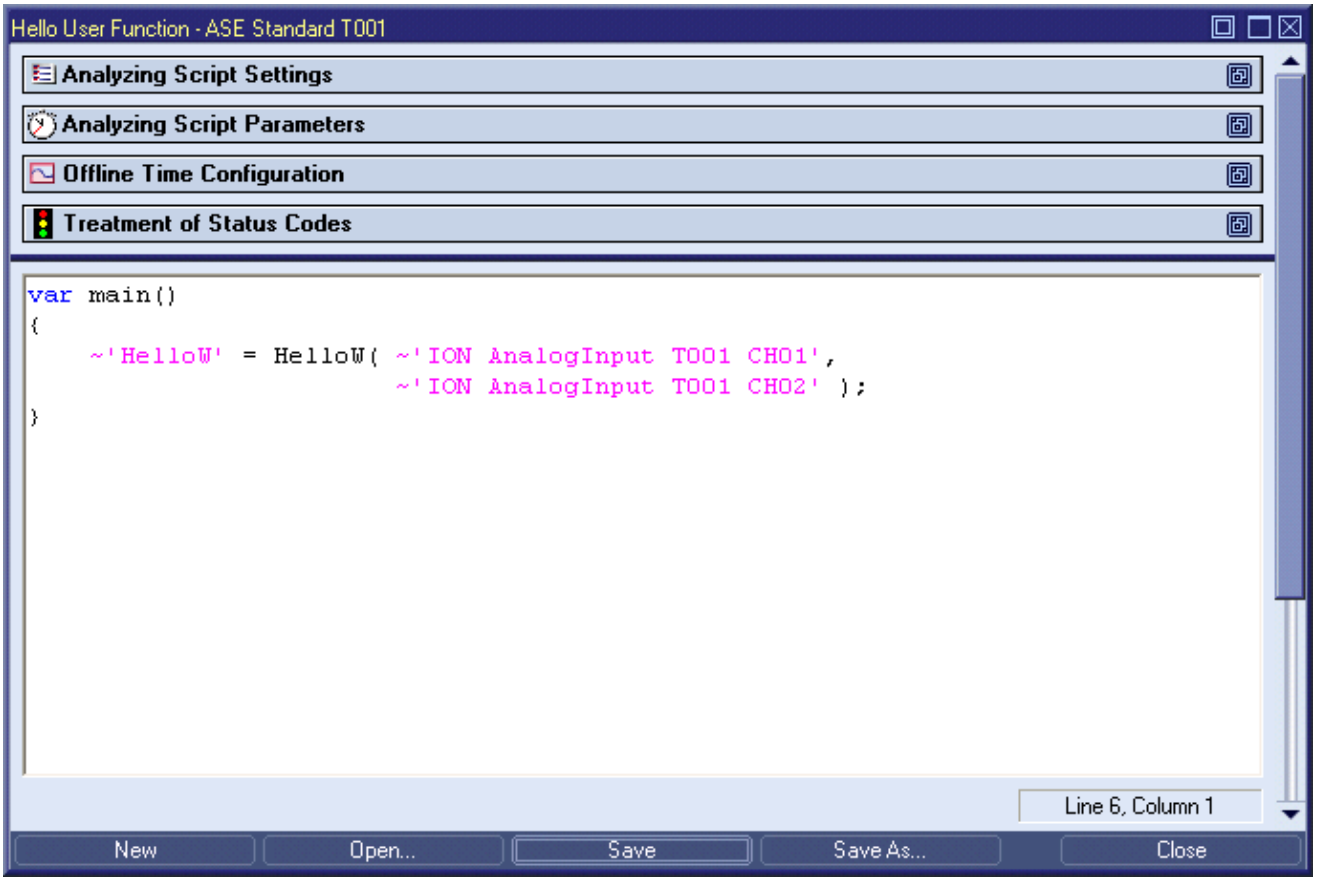


*Figure 45: **ASE Standard T001** with the HelloW() User Function*

➔ Start the Analyzing Script.

The output data which has been configured within the started Analyzing Script becomes available within the **MDS Explorer**.

➔ Create another new, empty Analyzing Script.

➔ Drag&Drop the User Function “HelloW2”, which has been created during the previous tutorial, from the **ANS Explorer** into the **ASE Standard T001**.

➔ Provide two online data from the **MDS Explorer** as first two parameters for HelloW() and specify a data name for the to be returned data as third parameter.

➔ The Analyzing Script looks like in the following example:

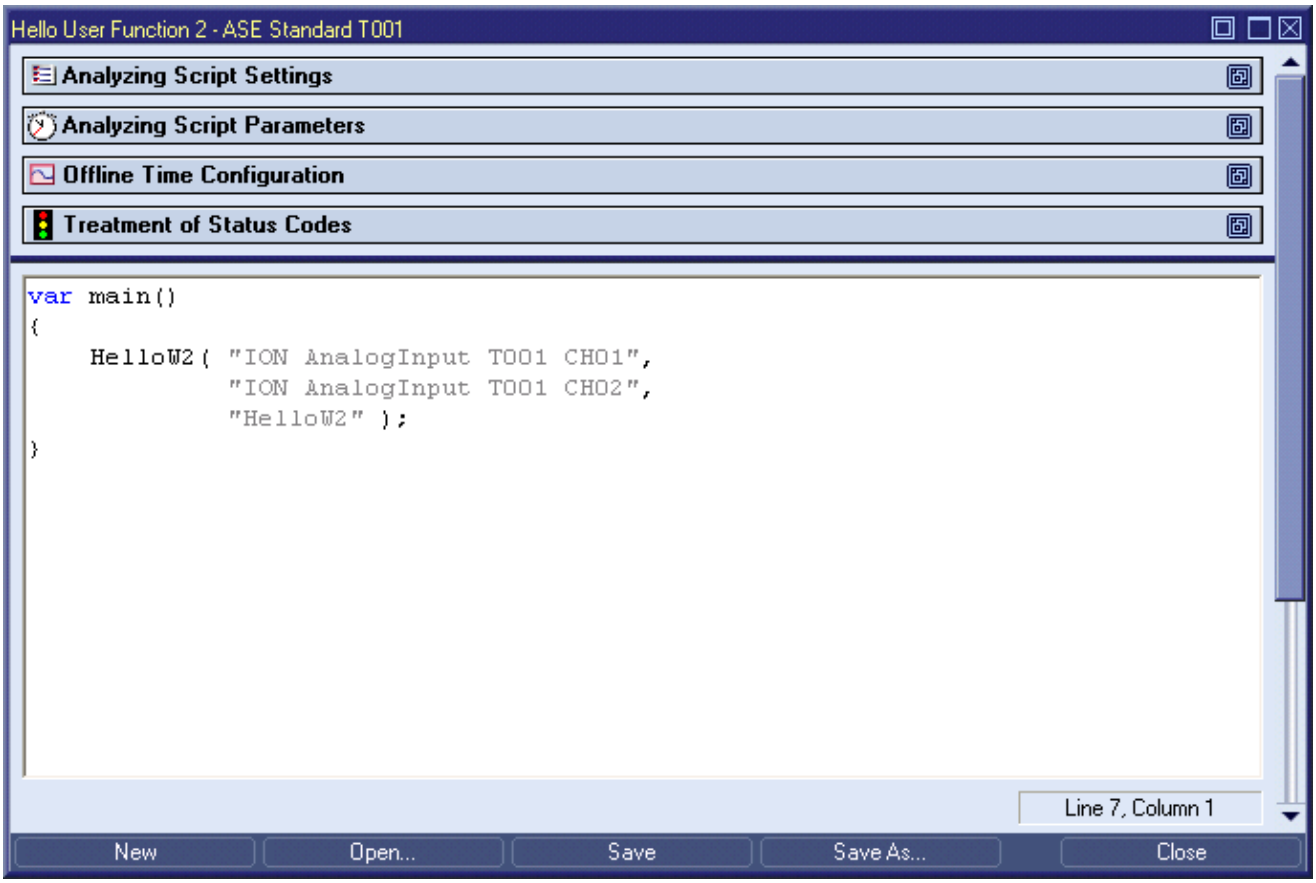


*Figure 46: **ASE Standard T001** with the HelloW2() User Function*

➔ Start the Analyzing Script.

The output data which has been configured within the started Analyzing Script becomes available within the **MDS Explorer**.

➔ Compare the output of both Analyzing Scripts within the Monitoring System.

Both Analyzing Script produce identical output data.

➔ Stop both Analyzing Scripts.

The data which have been configured within the started Analyzing Scripts are being removed from the **MDS Explorer**.

Ready

At this point you know how to use a User Function within an Analyzing Script.

6.15.6 Behavior of User Functions

The following list provides information about the special requirements and functionalities of User Functions:

- User Functions can be protected against reading.
  - In case a password has been specified for a User Function, the User Function can not be opened within the **UFE Standard T001** without entering of the proper password.
  - The password functionality of User Functions is independent to the user account which has been used for the login to the ***X-Tools Server***.
  - Protected User Functions are stored to disk with encryption, thus the contents of protected User Functions also can not be read outside of ***X-Tools***.
- Analyzing Models can be converted into User Functions and vice versa.
  - Within the **AME Standard T001**, the Convert button can be pressed in order to convert the currently opened Analyzing Model into a User Function. The default values for new User Functions are being used for all parameters and settings of the new User Function.
  - Within the **UFE Standard T001**, the Convert button can be pressed in order to convert the currently opened User Function into an Analyzing Model. The default values for new Analyzing Models are being used for all parameters and settings of the new Analyzing Model.
- User Function Libraries
  - Each User Function which is present within any of the currently present Configuration File locations also is shown below the “Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch. From there, the User Functions can be edited and they can be dropped into Analyzing Models and Analyzing Scripts.
  - Each User Function is displayed below the category which has been specified within the User Function Properties table of the **UFE Standard T001**.
- The short and full names of each User Function must be unique.
  - In case there is an Analyzing Function with the same name as a User Function, the Analyzing Function is being used for the calculation and the User Function is being ignored.
  - In case there are two User Functions of identical name, none of them is being used for the calculation and the parent Analyzing Model or Analyzing Script suspends.
  - User Functions which have a naming conflict (either with an Analyzing Function or with another User Function) show this status via an additional icon below the “Main -> Servers -> *YourServer* -> Libraries -> User Functions” branch.
  - In order to resolve the naming conflict of a User Function, the according User Function must be opened, the short and/or full names must be changed and the User Function must be saved afterwards.

6.15.7 Examples of User Functions

The example location which is delivered together with ***X-Tools*** contains a few examples of User Functions. For each User Function there are also examples of Analyzing Models and Analyzing Scripts available:

- ConvI16ToB
  - Description
    - Shows how to convert a single 16-bit Integer into 16 single bits.
  - Example Analyzing Models
    - Conversion\ConvI16ToB 01 (online data)
    - Conversion\ConvI16ToB 02 (offline data)
  - Example Analyzing Scripts
    - Conversion\ConvI16ToB 01 (online data)
    - Conversion\ConvI16ToB 02 (offline data)
  - Example Monitoring Views
    - Example for ConvI16ToB()
      - This Monitoring View shows the online and offline output of the Analyzing Models and Analyzing Scripts.
- MyFFT
  - Description
  - Shows how to create a User Function which embeds all functionality for calculating of standardized FFTs.
  - Example Analyzing Models
    - Spectral\MyFFT 01 (online data)
  - Example Analyzing Scripts
    - Spectral\MyFFT 01 (online data)
- MyRMS
  - Description
  - Example Analyzing Models
    - Statistic\MyRMS 01 (online data)
  - Example Analyzing Scripts
    - Statistic\MyRMS 01 (online data)

6.16 Storage of offline Data via Storage Profiles

6.16.1 General Description

In this chapter you will learn how to store offline data from the **MDS Explorer** with the **Storage System** via Storage Profiles for offline data. A Storage Profile for offline data contains all of the information which is needed in order to perform the storage of offline data to a storage location. This information includes, among others, the names of the to-be-stored data, paths, trigger settings and clean-up settings. Multiple Storage Profiles can be started and running simultaneously at any time.

The offline data which is being stored by a running Storage Profile can be loaded as offline data afterwards via an **Offline Data Loader**.

Examples of Storage Profiles are found within the example Configuration File location which is being delivered together with ***X-Tools***. See also point 6.9 for information about how to store online data via Storage Profiles. In addition, point 6.9 also shows how to open existing Storage Profiles and how to start and stop Storage Profiles - as these operations are equal for all Storage Profiles, these operations are not being explained within this chapter again.

Whenever a Storage Profile with offline data as input runs into an error, it stops automatically. This is different to Storage Profiles with online data as input, which suspend in such a situation and attempt to resume after the error condition is gone. The main reasons for this different kind of handling are:

- for Storage Profiles with offline data as input it is much less likely that the error condition will disappear ever
- a main use cases for Storage Profiles with offline data as input are sequential Analyzing Scripts (see point 6.20), which could be blocked by a suspended Storage Profile without any benefit

This chapter contains the following topics:

- **Creation of a new Storage Profile for offline Data**

6.16.2 Creation of a new Storage Profile for offline Data

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branch of the **MDS Explorer**.

Functionality

- Method 1 for the creation of a new Storage Profile:
  - Within the **STS Explorer**, expand the “Main -> Editors -> Storage Profile Editors” branch.
  - Drag the **SPE Binary T001** with the left mouse into the empty space at the right side of the **STS Explorer**. An empty **SPE Binary T001** is opened automatically.
  - Press the New button in order to create a new Storage Profile. The new Storage Profile is displayed by the **SPE Binary T001** with default values.
  - This method works in all cases, even in case there is no ***X-Tools Server*** connected at the moment. It can be used for the creation and configuration of Storage Profiles for cases where the target ***X-Tools Server*** is not present at the moment.

- Method 2 for the creation of a new Storage Profile:
  - Within the **STS Explorer**, expand the “Main -> Servers -> *YourServer* -> Libraries -> Storage Process Modules” branch.
  - Click onto the present STE SPM Binary T001 with the right mouse button in order to open the context menu.
  - In order to create a new Storage Profile for the storage format “Binary T001”, choose New Storage Profile from the context menu.
  - This method works only in case the target ***X-Tools Server*** is present but it is more convenient and faster as it automatically also determines the target ***X-Tools Server*** of the new Storage Profile.
- Method 3 for the creation of a new Storage Profile:
  - Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branch.
  - Drag&Drop the desired target data into the empty **Storage System**. The **SPE Binary T001** is opened automatically, a new Storage Profile is being initialized with default values and the dropped data is added to Data table. As offline data is being dropped, the Time Domain cell within the Storage Profile Parameters table is being set to “Offline Data with absolute Timestamps” or “Offline data with relative Timestamps” automatically.
  - This method works only in case the target ***X-Tools Server*** and the target data are present but it is more convenient and faster as it automatically opens all windows.
- The following screenshot shows an example of a new Storage Profile.

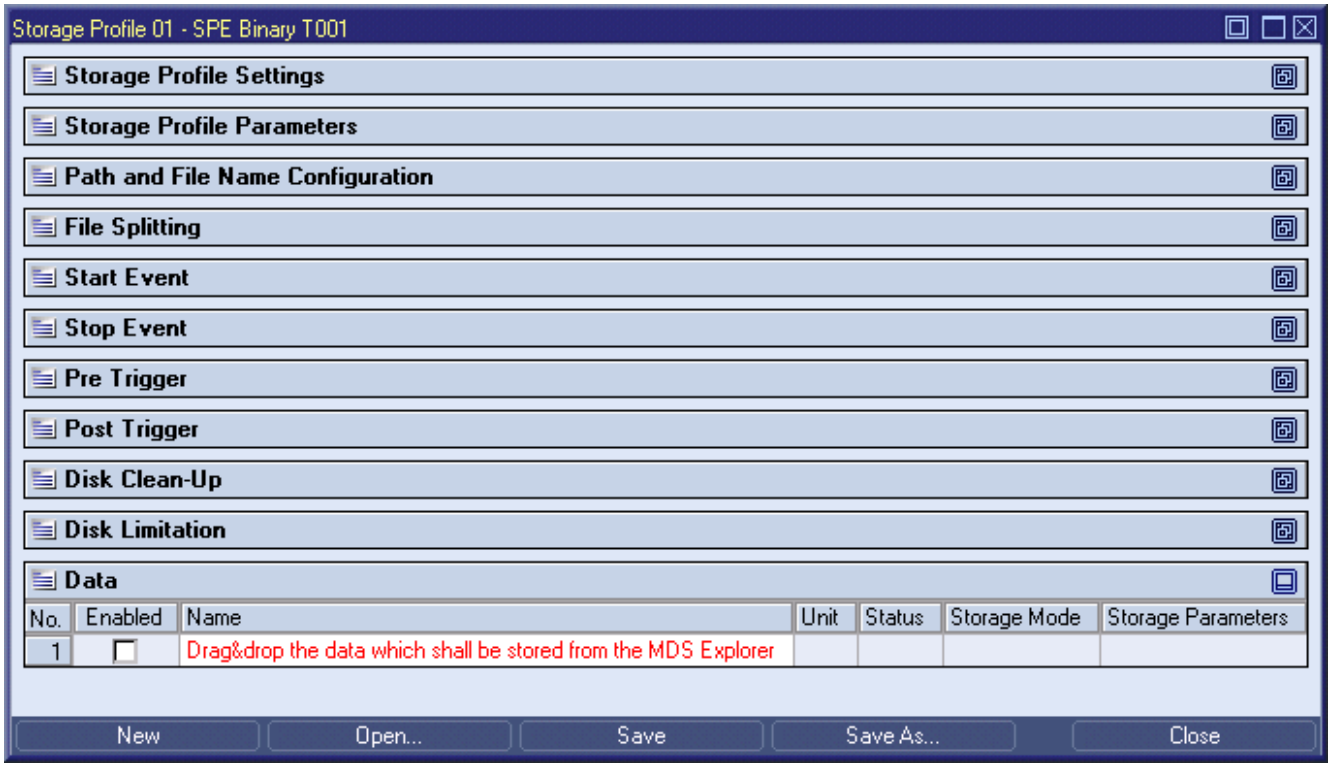


*Figure 47: **SPE Binary T001** with a new Storage Profile*

- See point 6.9 for detailed information about the provided controls.
- For a more detailed description of all available functionalities of the **SPE Binary T001**, refer to the reference manual of the “Storage System”, chapter “SPE Binary T001”.

Try it out

➔ Create a new Storage Profile.

➔ Specify the desired data storage location.

➔ Change the file splitting interval to 1 minute.

➔ Add some data names to the Data table. Specify different storage modes for each of the to-be-stored data.

➔ Save the Storage Profile. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to “Hello Storage Profile for offline Data” and press OK in order to save it to the disk.

The new Storage Profile now is displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Storage Profile Files” branch of the **STS Explorer**.

➔ Take a look onto the Storage Profiles which are provided via the example location (see point 6.5) in order to see various application examples about the capabilities and usage of Storage Profiles.

Ready

At this point you know how to create a new Storage Profile which can be used in order to store offline data to the disk.

6.17 Scalar Data

6.17.1 General Description

In this chapter you will learn how to create and how to use scalar data. In case it becomes necessary to edit and/or to prepare scalar data outside of ***X-Tools***, refer to chapter 7.4.20.

This chapter contains the following topics:

- **Creation of scalar Data via the MDS Explorer**
- **Creation of scalar Data via Analyzing Models and Analyzing Scripts**
- **Using of scalar Data within Analyzing Models and Analyzing Scripts**
- **Using of scalar Data for Input Arguments of Class “Parameter”**
- **Changing of the Value of a scalar Data via the MTC Mimic Board T001**

6.17.2 Creation of scalar Data via the MDS Explorer

Functionality

- Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer* -> Scalar Data” branch.
- Call the context menu and choose Add Scalar Data….
  - The **Add Scalar Data** dialog opens and allows to enter the properties of the new scalar data.
- Enter reasonable values for all properties of the new scalar data and press Add.
  - The new scalar data appears within the **MDS Explorer** below the “Main -> Servers -> *YourServer* -> Scalar Data” branch.

Try it out

➔ Add a few scalar data of different data types and with different values to the **MDS Explorer**.

Ready

At this point you know how to use the **Add Scalar Data** dialog in order to add scalar data to the **MDS Explorer**.

6.17.3 Creation of scalar Data via Analyzing Models and Analyzing Scripts

Functionality

- Create a new Analyzing Model with a cycle time of 1 second which contains the following Analyzing Functions and input arguments:

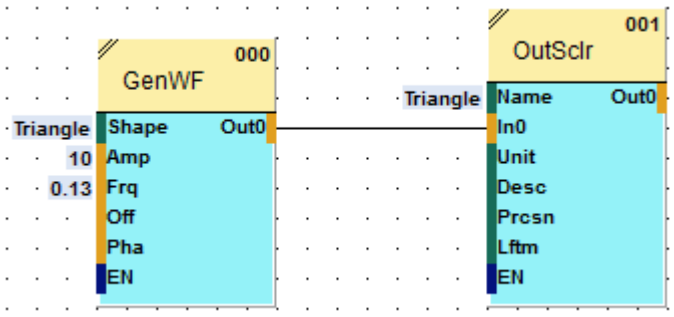


Figure 48: The Analyzing Model "Tutorial\Scalar Data\Triangle" from the Example Location

- An example for this Analyzing Model also is found within the "..\Example Location\Analyzing Model Files\Tutorial\Scalar Data" directory which is delivered together with ***X-Tools***.
- Save the Analyzing Model and start it.
  - The new scalar data "Triangle" appears within the **MDS Explorer** below the "Main -> Servers -> *Your-Server* -> Scalar Data" branch.
- Create a new Analyzing Script with a cycle time of 1.123 seconds which contains the following Analyzing Functions and input arguments:

```
var main()
{
    s'Triangle' = GenWF( "Triangle", 20, 0.37 );
}
```

Figure 49: The Analyzing Script "Tutorial\Scalar Data\Triangle" from the Example Location

- An example for this Analyzing Script also is found within the "..\Example Location\Analyzing Script Files\Tutorial\Scalar Data" directory which is delivered together with ***X-Tools***.
  - Instead of the s'' syntax, also the Analyzing Function OutSclr() can be used within Analyzing Scripts. This works in the same way like the ~'' syntax for online data and the $'' syntax for offline data.
- Save the Analyzing Script and start it.
  - At this moment, the Analyzing Model and the Analyzing Script write the value and properties of the same scalar data. The scalar data always remembers the properties and the value which have been written last.
- The following steps show better results when the value of the used scalar data is changed only by one writer. It is suggested (but not mandatory) to stop either the Analyzing Model or the Analyzing Script before the next steps are processed.

Try it out

➔ Create Analyzing Models which use OutSclr() and OutSclrStr() in order to add scalar data to the **MDS Explorer**.

➔ Create Analyzing Script which use OutSclr() and OutSclrStr() in order to add scalar data to the **MDS Explorer**.

Ready

At this point you know how to use Analyzing Models and Analyzing Scripts in order to add scalar data to the **MDS Explorer**.

6.17.4 Using of scalar Data within Analyzing Models and Analyzing Scripts

Functionality

- Create a new Analyzing Model with the default cycle time of 1 milli second which contains the following Analyzing Functions and input arguments:

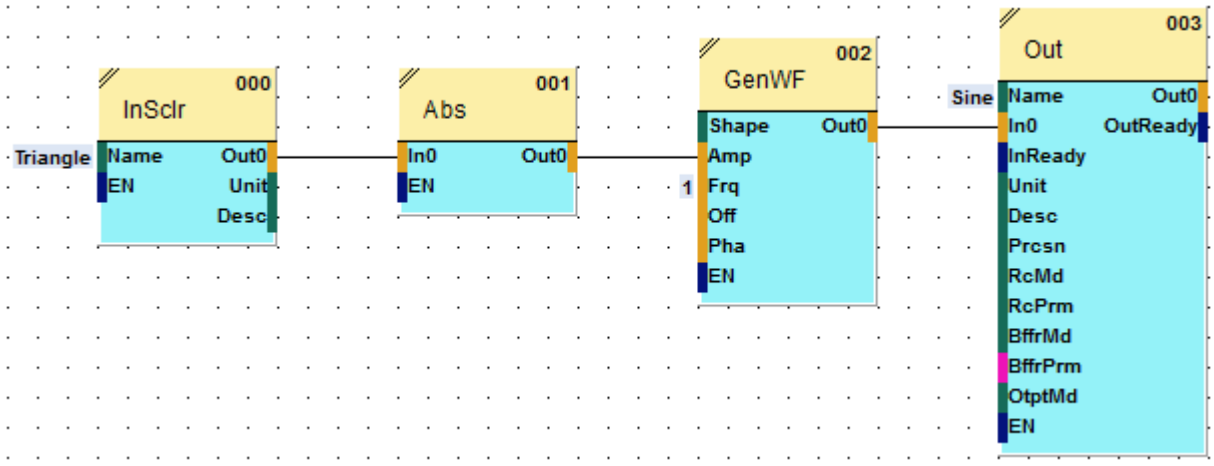


Figure 50: The Analyzing Model "Tutorial\Scalar Data\Sine" from the Example Location

- An example for this Analyzing Model also is found within the "..\Example Location\Analyzing Model Files\Tutorial\Scalar Data" directory which is delivered together with ***X-Tools***.
- Save the Analyzing Model and start it.
  - The new online data "Sine" appears within the **MDS Explorer** below the "Main -> Servers -> *Your-Server* -> Online Data -> All Online Data" branch.
  - The maximal amplitude of the output online data "Sine" is defined through the current value of the scalar data "Triangle".
- Visualize the values of the scalar data "Triangle" and of the online data "Sine" within the Monitoring System.
  - An example Monitoring View also is found within the "..\Example Location\Monitoring View Files\Tutorla\Scalar Data" directory which is delivered together with ***X-Tools***.

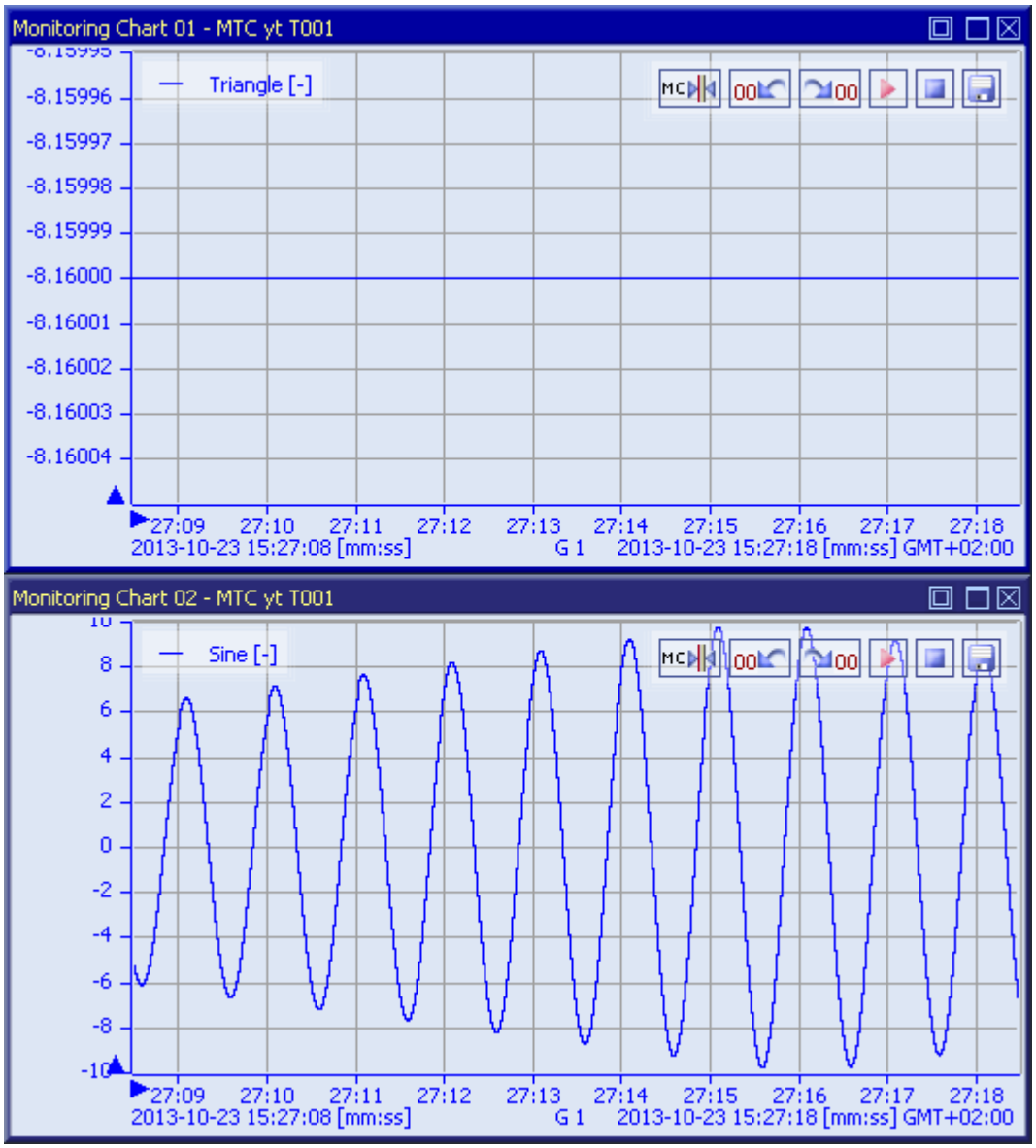


Figure 51: The Monitoring View "Tutorial\Scalar Data\Triangle + Sine" from the Example Location

- Create a new Analyzing Script with the default cycle time of 1 milli seconds which contains the following Analyzing Functions and input arguments:

```
var main()
{
    ~'Multiplication' = 15 * s'Triangle';
}
```

Figure 52: The Analyzing Script "Tutorial\Scalar Data\Multiplication" from the Example Location

- An example for this Analyzing Script also is found within the "..\Example Location\Analyzing Script Files\Tutorial\Scalar Data" directory which is delivered together with ***X-Tools***.
    - Instead of the s'' syntax, also the Analyzing Function InSclr() can be used within Analyzing Scripts. This works in the same way like the ~'' syntax for online data and the $'' syntax for offline data.
- Save the Analyzing Script and start it.
    - The new online data "Multiplication" appears within the **MDS Explorer** below the "Main -> Servers -> *YourServer* -> Online Data -> All Online Data" branch.
    - The value of the output online data "Multiplication" is defined through the current value of the scalar data "Triangle" which is multiplied by 15.

- Visualize the values of the scalar data “Triangle” and of the online data “Multiplication” within the Monitoring System.
  - An example Monitoring View also is found within the “..\Example Location\Monitoring View Files\Tutorla\Scalar Data” directory which is delivered together with ***X-Tools***.

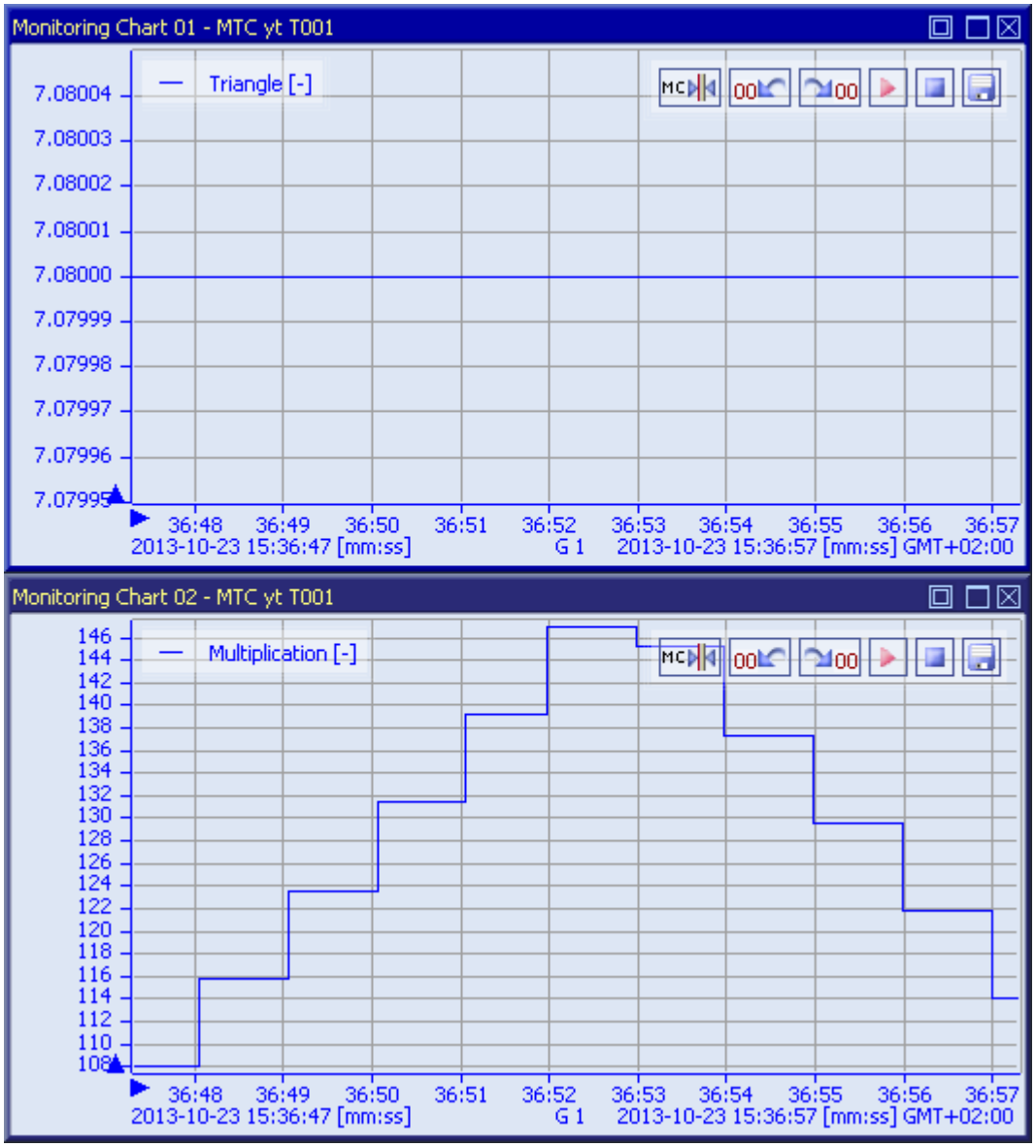


Figure 53: The Monitoring View “Tutorial\Scalar Data\Triangle + Multiplication” from the Example Location

Try it out

➔ Create Analyzing Models which use InSclr() and InSclrStr() in order to read the current values of scalar data.

➔ Create Analyzing Scripts which use InSclr(), InSclrStr() and the s’’ syntax in order to read the current values of scalar data.

Ready

At this point you know how to use Analyzing Models and Analyzing Scripts in order to read the current values from scalar data.

6.17.5 Using of Scalar Data for Input Arguments of Class “Parameter”

Functionality

- Start the Analyzing Script “Tutorial\Scalar Data\Create Parameters” from the example location which is delivered together with ***X-Tools***.
  - This Analyzing Script adds 6 scalar data to the **MDS Explorer**.

```
var main()
{
    OutSclr(    "Spectrum.Input.Amplitude",      10,                  "n/a", "amplitude of the input data",         "Decimal32", "permanent" );
    OutSclr(    "Spectrum.Input.Frequency",       5,                  "Hz",  "frequency of the input data",         "Decimal32", "permanent" );
    OutSclr(    "Spectrum.Calculation.NoP",    2048,                  "NoP", "number of points of the calculation", "Integer16", "permanent" );
    OutSclrStr( "Spectrum.Output.Name",        "Spectrum",        16, "n/a", "name of the output data",                         "permanent" );
    OutSclrStr( "Spectrum.Output.Unit",        "Some Unit",       64, "n/a", "unit of the output data",                         "permanent" );
    OutSclrStr( "Spectrum.Output.Description", "Some Description", 64, "n/a", "description of the output data",                 "permanent" );

}
```

Figure 54: The Analyzing Script “Tutorial\Scalar Data\Create Parameters” from the Example Location

- Start the Analyzing Model “Tutorial\Scalar Data\Use Parameters” from the example location which is delivered together with ***X-Tools***.
  - This Analyzing Model uses the 6 scalar data from the Analyzing Script in order to define all of the parameters for its input, its calculation and its output. As scalar data also can be used for setting of the value of input arguments of class “parameter”, they provide a convenient way to change the value of multiple input arguments at a central place.
  - The new online data “Spectrum” appears within the **MDS Explorer** below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch.

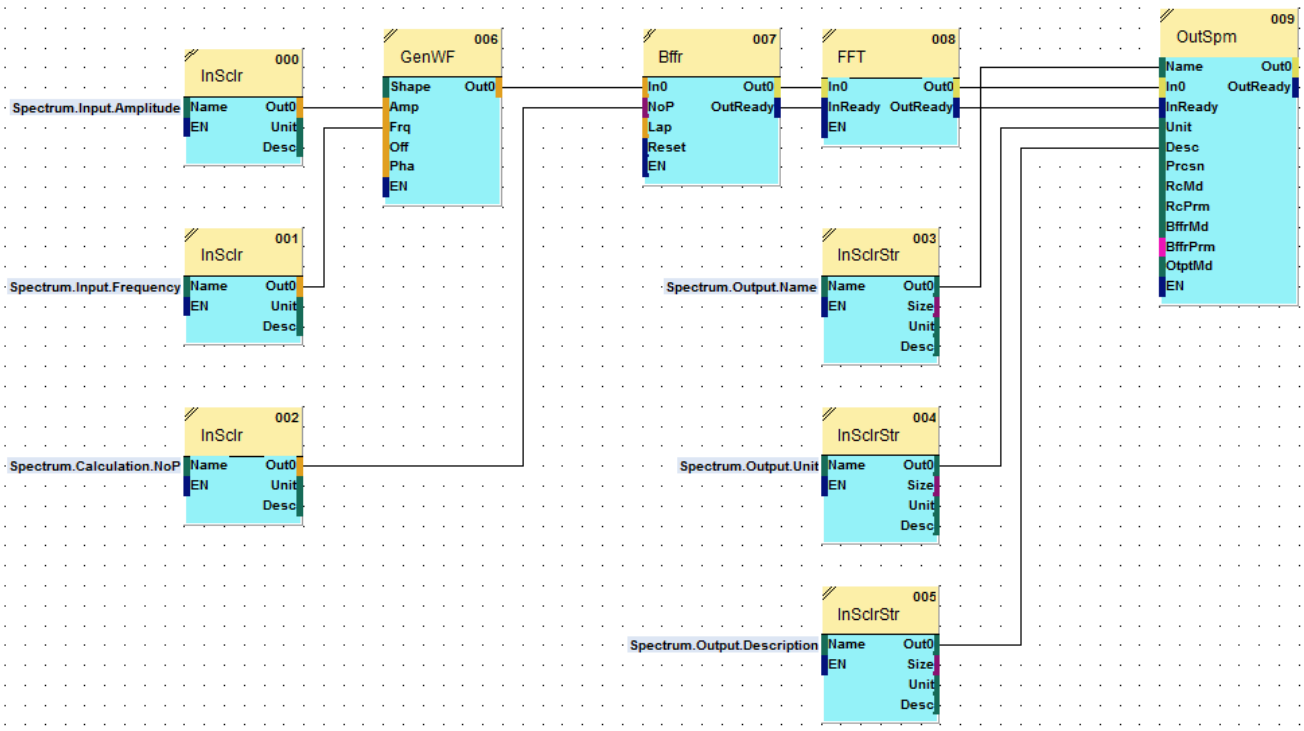


Figure 55: The Analyzing Model “Tutorial\Scalar Data\Use Parameters” from the Example Location

- In addition to the **MTC yt T001**, also the **MTC Mimic Board T001** and the **MTC yn T001** can display the values of scalar data. The following screenshot shows an example Monitoring View which uses the scalar data value “Sine” as the data which is linked to the Main Frequency Marker of a **MTC yn T001**. In addition, a **MTC Mimic Board T001** is used in order to output the values of scalar data of type of Integer, decimal and string.

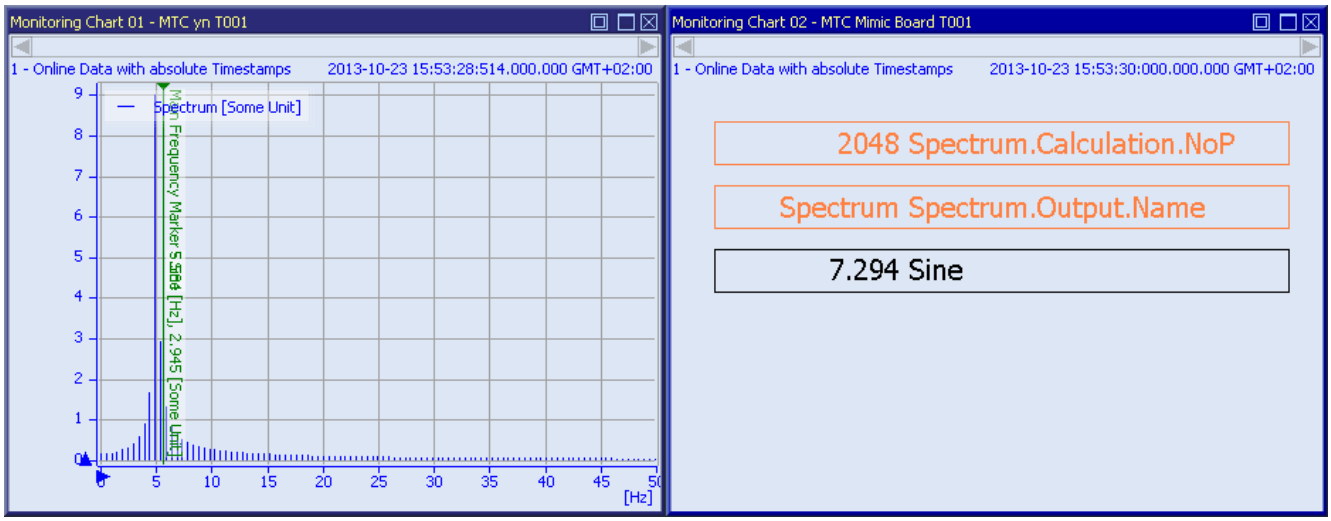

Monitoring Chart 01 - MTC yn T001
1 - Online Data with absolute Timestamps
2013-10-23 15:53:28:514.000.000 GMT+02:00
Spectrum [Some Unit]
Frequency Marker 5.294 [Hz], 2.945 [Some Unit]
[Hz]
Monitoring Chart 02 - MTC Mimic Board T001
1 - Online Data with absolute Timestamps
2013-10-23 15:53:30:000.000.000 GMT+02:00
2048 Spectrum.Calculation.NoP
Spectrum Spectrum.Output.Name
7.294 Sine

Figure 56: The Monitoring View "Tutorial\Scalar Data\Spectrum" from the Example Location

Try it out

➔ Create Analyzing Models which use InSclr() and InSclrStr() in order to read the current values of scalar data in order to use them for the initialization of their input arguments of class "parameter".

Ready

At this point you know how to use scalar data and Analyzing Models in order to initialize input arguments of class "parameter" with the values of scalar data.

6.17.6 Changing of the Value of a scalar Data via the MTC Mimic Board T001

Functionality

- Stop all Analyzing Models and Analyzing Scripts.
  - Nobody is writing the values of scalar data now, but all scalar data remain present within the **MDS Explorer** and remember their last value.
- Open a new **MTC Mimic Board T001** and drag a few scalar data into it.
  - A value box is inserted automatically for each scalar data.
  - Each value box displays the current value of its scalar data.
- Switch the **MTC Mimic Board T001** to the view mode via the leftmost toolbar button.
- Double-click onto a value box in order to open the **Edit Value Box** dialog.
  - The only parameter which can be changed within this dialog is the value of the scalar data.
  - Change the value of the scalar data to a new value and press OK.
  - After the change, the new value of the scalar data is displayed within the Monitoring System, within the **Edit Scalar Data** dialog of the **MDS Explorer** and within the **Edit Value Box** dialog.

Try it out

➔ Drag all 6 scalar data from the "Spectrum" category into a new **MTC Mimic Board T001**.

➔ Change the values of all scalar data via the **MTC Mimic Board T001**.

➔ Start the Analyzing Model "Tutorial\Scalar Data\Use Parameters" from the example location which is delivered together with ***X-Tools***.

  - The Analyzing Model starts and runs with the values that have been configured via the **MTC Mimic Board T001**.

Ready

At this point you know how to use the **MTC Mimic Board T001** in order to configure the values of scalar data which afterwards can be used for the input values of Analyzing Functions.

6.18 3D Visualization

6.18.1 General Description

In this chapter you will learn how to visualize 2-dimensional data from the **MDS Explorer** with the 3D functionality of the **Monitoring System**. Several Analyzing Functions provide the results of their calculations as 2-dimensional data. In order to visualize 2-dimensional data over the time, the 3D functionality of the **Monitoring System** is being used.

This chapter contains the following topics:

- **Visualization of Data in 3D**

6.18.2 Visualization of Data in 3D

Provide some 2-dimensional Data

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.
- Ensure that the example location is present as one of the current locations (see point 6.5.2 for how to add an additional location).
- From the example location, start the Analyzing Model “H2D 01” which is found below the sub-folder “Statistic”. After this Analyzing Model has been started, the online data “H2DForSignal” appears within the **MDS Explorer**.

Functionality

- Open the **Monitoring System**.
- Within the **MDS Explorer**, expand the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” branch.
- Drag&Drop a 2-dimensional data into the empty **Monitoring System**. The **MVE Standard T001** is opened automatically and also opens the **MTC ynm T001**. In addition, the dropped data is added to and visualized by the opened **Monitoring Chart** automatically.

- The following screenshot shows an example of a 3D visualization:

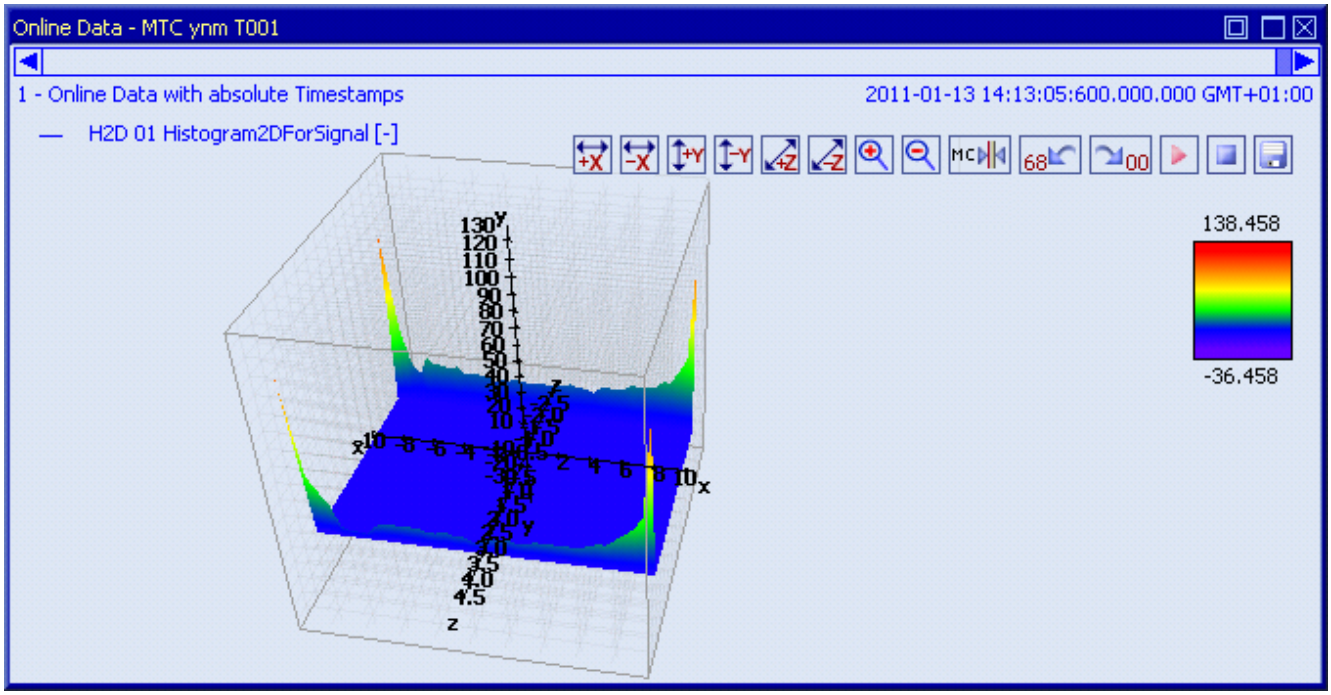


*Figure 57: **MVE Standard T001** with a 3D Visualization*

- Several mechanisms are available for the changing of the visualization of the displayed data:
  - When the left mouse button is being pressed, the displayed cuboid can be rotated.
  - When the right mouse button is being pressed, the displayed cuboid can be moved.
- Via the tool bar in the right corner of each **Monitoring Chart**, basic functionalities are available:
  - each axis of the cuboid can be stretched and shrunken independently
  - all axes of the cuboid can be stretched and shrunken simultaneously
  - the cursors can be turned on/off
  - the last zooming, shifting and scrolling operations can be undone/redone
  - the visualization can be paused/continued

Try it out

➔ Create a new Monitoring View.

➔ Drag some 2-dimensional online data from the **MDS Explorer** into the Monitoring View in order to visualize the data curves.

➔ Pause the visualization, turn the cursors on and take a look onto the displayed measurement values. Continue the visualization, zoom into the visualization, shrink and stretch the axes scaling, rotate and move the cuboid and use the undo/redo buttons. Repeat these operations in random order in order to adjust the visualization to the needs of the current application.

Ready

At this point you know how to visualize 2-dimensional data with the 3D functionality of the **Monitoring System**.

6.19 Safe Mode

6.19.1 General Description

In this chapter you will learn how to use the safe mode of the ***X-Tools Server*** in order to recover automatically from severe error conditions and in order to allow the ***X-Tools Server*** to continue with its measurement, analysis and storage tasks afterwards.

This chapter contains the following topics:

- **Important Facts**
- **Safe Mode**
- **X-Tools Service**

6.19.2 Important Facts

During the startup of the ***X-Tools Server***, each operation (e.g. initialization of a component or start of a Configuration File, …) is logged into an according Initialization Log File. When a severe error (e.g. a crash or a hang, …) is being detected and the ***X-Tools Server*** is being restarted in safe mode afterwards, the formerly created Initialization Log File is being examined and the operation which probably caused the previous error is not being executed again. Thus, the remaining functionality which has been configured can be started and executed by the ***X-Tools Server***.

To sum up, the safe mode detects severe error conditions, shall reduce the downtime and shall keep the application running.

6.19.3 Safe Mode

In order to allow the ***X-Tools Server*** to startup in safe mode it must be started with the "/safe" command line parameter. Even in case the "/safe" command line parameter is being provided, the ***X-Tools Server*** starts up in safe mode only in case the Initialization Log File indicates an error during the previous execution. Typically, there is no need to start the ***X-Tools Server*** manually because it is started automatically either by the ***X-Tools Client*** or during the startup of Windows. The ***X-Tools Client*** starts the ***X-Tools Server*** without the "/safe" command line parameter, whereas the "/safe" command line parameter is being used whenever the ***X-Tools Server*** is being started automatically during the startup of Windows or by the ***X-Tools Service***.

In case the ***X-Tools Server*** starts up in safe mode, an artificial delay of 30 seconds is inserted between the starts of any two Configuration Files. In addition, according log messages are added to the System Log during the startup in safe mode.

In case a severe error has been detected and the ***X-Tools Server*** is restarted in safe mode afterwards, an email notification can be sent to the configured email recipients (see **Global Options Editor**, refer to the reference manual of the "Main Management System", chapter "Global Options Editor").

The icon of an ***X-Tools Server*** which is running in safe mode indicates this special operating condition:

Icon	Description
(normal operation)	The ***X-Tools Server*** is running normally.
(safe mode)	The ***X-Tools Server*** is running in safe mode.

6.19.4 X-Tools Service

In case the ***X-Tools Server*** crashes or hangs it must be restarted by another application. The ***X-Tools Service*** takes over this task and must be enabled in order to enable the restart functionality. In order to enable the re-start through the ***X-Tools Service***, choose “Automatically restart the X-Tools Server in case it stops to operate normally” for the Automated Restart setting within the Global Options of the ***X-Tools Server***.

When it is enabled, the ***X-Tools Service*** constantly monitors the ***X-Tools Server*** and restarts it automatically in case of a detected crash or hung. In addition, the ***X-Tools Service*** itself is monitored by Windows and Windows restarts the ***X-Tools Service*** automatically in case it disappears. In case the context menu of the ***X-Tools Server*** (from the task bar of Windows) is not available for any reason and the ***X-Tools Server*** must be stopped, the ***X-Tools Service*** must be stopped earlier via the “Services” dialog of Windows which is found in “Start” -> “Control Panel” -> Administrative Tools” -> “Services”. If the ***X-Tools Service*** would not be stopped and the ***X-Tools Server*** would be killed (e.g. via the Task Manager), it would be restarted automatically by the ***X-Tools Service***.

6.20 Sequence Control

6.20.1 General Description

In this chapter you will learn how to use the sequence control in order to automatically start and stop any measurement and analysis tasks. The sequence control is configured within an Analyzing Script and is being used especially for applications where the to-be-performed analysis tasks exceed the calculation possibilities of the used computer. In such a case, multiple analysis tasks can be performed after each other instead of parallel, which allows more complex analysis tasks.

This chapter contains the following topics:

- **Important Facts**
- **Provided Analyzing Functions**
- **Example**

6.20.2 Important Facts

In order to configure a sequence control of measurement and analysis tasks, a simple Analyzing Script is being used. The Analyzing Script calls Analyzing Functions in order to start and stop all of the needed Configuration Files. An example use case could be:

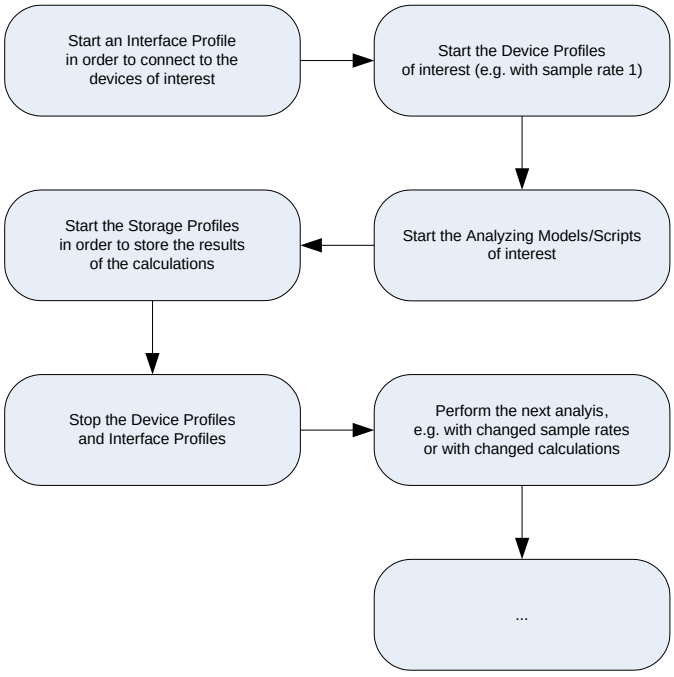

Start an Interface Profile
in order to connect to the
devices of interest
Start the Device Profiles
of interest (e.g. with sample rate 1)
Start the Storage Profiles
in order to store the results
of the calculations
Start the Analyzing Models/Scripts
of interest
Stop the Device Profiles
and Interface Profiles
Perform the next analyis,
e.g. with changed sample rates
or with changed calculations
...

Figure 58: Example Use Case for the Sequence Control

6.20.3 Provided Analyzing Functions

The following Analyzing Functions are being used for the execution of the sequence control. Refer to the reference manual of the “Analyzing System”, chapter “Analyzing Functions” for detailed information about each of these Analyzing Functions.

Start()

- allows to start a Configuration File (an Interface Profile, a Device Profile, an Analyzing Model, an Analyzing Script or a Storage Profile)
- via the “WaitMode” parameter it can be configured whether Start() shall wait until the specified Configuration File has finished its operation or not
  - typically it does not make sense to wait for Interface Profiles and Device Profiles or for Analyzing Models, Analyzing Scripts and Storage Profiles in case they are operating on online data
  - in case of Analyzing Models, Analyzing Scripts and Storage Profiles which are operating on offline data, it can make sense to wait until the execution (e.g. the calculation or the storage) has finished
- the “Start 01” Analyzing Script within the example location shows another example about how to use this Analyzing Function

Pause()

- allows to pause a Configuration File (an Interface Profile, a Device Profile, an Analyzing Model, an Analyzing Script or a Storage Profile)
- the “Pause 01” Analyzing Script within the example location shows another example about how to use this Analyzing Function

Continue()

- allows to continue a Configuration File (an Interface Profile, a Device Profile, an Analyzing Model, an Analyzing Script or a Storage Profile)
- via the “WaitMode” parameter it can be configured whether Continue() shall wait until the specified Configuration File has finished its operation or not
  - typically it does not make sense to wait for Interface Profiles and Device Profiles or for Analyzing Models, Analyzing Scripts and Storage Profiles in case they are operating on online data
  - in case of Analyzing Models, Analyzing Scripts and Storage Profiles which are operating on offline data, it can make sense to wait until the execution (e.g. the calculation or the storage) has finished
- the “Pause 01” Analyzing Script within the example location shows another example about how to use this Analyzing Function

Stop()

- allows to stop a Configuration File (an Interface Profile, a Device Profile, an Analyzing Model, an Analyzing Script or a Storage Profile)
- the “Stop 01” Analyzing Script within the example location shows another example about how to use this Analyzing Function

RemoveData()

- in case an Analyzing Model or Analyzing Script is operating on offline data, the calculated results remain below “Main -> Servers -> YourServer >Offline Data -> All Offline Data” also after the Analyzing Model/Script has been stopped so that other systems (e.g. following Analyzing Models/Scripts or following Storage Profiles) can continue to process these results
- in order to remove previously created offline data automatically, RemoveData() is being used
- the “RemoveData 01” Analyzing Script within the example location shows another example about how to use this Analyzing Function

WaitForCF()

- allows to wait until a Configuration File has reached the specified state
- this Analyzing Function is being used especially in cases where multiple Analyzing Models/Scripts and/or Storage Profiles have to run in parallel, because it allows to use Start() in order to start these multiple Configuration Files first and to wait until the last of them has reached the state of interest afterwards
- the “WaitForCF 01” Analyzing Script within the example location shows another example about how to use this Analyzing Function

WaitForEDT()

- allows to wait until a certain date and time has been reached
- this Analyzing Function can be used in order to perform measurement and analysis tasks periodically, e.g. once each hour or once each day
- the “WaitForEDT 01” Analyzing Script within the example location shows another example about how to use this Analyzing Function

IsDataPrsnt()

- allows to check whether the specified data is currently present within ***X-Tools***
- this Analyzing Function can be used in order to verify the execution during the processing of a measurement and analyzing task
- it also can be used in order to wait until a certain data becomes available, e.g. in case there are analyzing tasks which are depend from each other (one Analyzing Model/Script bases on the data which has been calculated by a previous Analyzing Model/Script)
- the “IsDataPrsnt 01” Analyzing Script within the example location shows another example about how to use this Analyzing Function

Exit()

- allows to stop the execution of the Analyzing Model or Analyzing Script which calls this Analyzing Function
- this Analyzing Function can be used in order to stop an Analyzing Model or Analyzing Script which calls it, e.g. in case a certain condition is reached
- the “Exit 01” and “Exit 02” Analyzing Scripts as well as the “Exit 01” and “Exit 02” Analyzing Models within the example location show other examples about how to use this Analyzing Function

6.20.4 Example

The “Sequence Control” directory of the Analyzing Scripts within the example location contains the Analyzing Script “Sequence Control 01”. This Analyzing Script shows different examples for the execution of automated measurement and analysis tasks:

- sequential processing of analyzing tasks:

```
var PerformSequentialCalculations()
{
    // start the Interface Profile which enables the simulated devices
    Start("://Simulation T001.ipf.xts", "DoNotWait");

    // start the Device Profiles of the simulated devices
    Start("://ION AnalogInput T001.dpf.xts", "DoNotWait");
    Start("://ION SIMATIC S7 PN.dpf.xts", "DoNotWait");

    // execute the desired calculations (one after the other)
    Start("://Sequence Control/Sequence Control 01.amf.xts", "Wait");
    Start("://Sequence Control/Sequence Control 02.asf.xts", "Wait");

    // store the calculated data
    Start("://Sequence Control 01.spf.xts", "Wait");

    // stop the Device Profiles
    Stop("://ION AnalogInput T001.dpf.xts");
    Stop("://ION SIMATIC S7 PN.dpf.xts");

    // stop the Interface Profile
    Stop("://Simulation T001.ipf.xts");

    // remove the calculated data
    RemoveData("://Sequence Control/Sequence Control 01.amf.xts");
    RemoveData("://Sequence Control/Sequence Control 02.asf.xts");
}
```

- parallel processing of analyzing tasks:

```
var PerformParallelCalculations()
{
    // start the Interface Profile which enables the simulated devices
    Start("://Simulation T001.ipf.xts", "DoNotWait");

    // start the Device Profiles of the simulated devices
    Start("://ION VIB-A T001.dpf.xts", "DoNotWait");
    Start("://IFN VIB-ACC.dpf.xts", "DoNotWait");

    // execute the desired calculations (in parallel)
    Start("://Sequence Control/Sequence Control 02.amf.xts", "DoNotWait");
    Start("://Sequence Control/Sequence Control 03.asf.xts", "DoNotWait");

    // wait until both calculations have finished
    WaitForCF("://Sequence Control/Sequence Control 02.amf.xts", "Stopped";);
    WaitForCF("://Sequence Control/Sequence Control 03.asf.xts", "Stopped";);

    // store the calculated data
    Start("://Sequence Control 02.spf.xts", "Wait");

    // stop the Device Profiles
    Stop("://ION VIB-A T001.dpf.xts");
    Stop("://IFN VIB-ACC.dpf.xts");

    // stop the Interface Profile
    Stop("://Simulation T001.ipf.xts");

    // remove the calculated data
    RemoveData("://Sequence Control/Sequence Control 02.amf.xts");
    RemoveData("://Sequence Control/Sequence Control 03.asf.xts");
}
```

- resolving of dependencies between analyzing tasks:

```
var PerformDependentCalculations()
{
    // start the Interface Profile which enables the simulated devices
    Start("://Simulation T001.ipf.xts", "DoNotWait");

    // start the Device Profiles of the simulated devices
    Start("://ION VIB-A T001.dpf.xts", "DoNotWait");
    Start("://IFN VIB-ACC.dpf.xts", "DoNotWait");

    // execute the desired calculations (in parallel)
    Start("://Sequence Control/Sequence Control 02.amf.xts", "DoNotWait");

    while(IsDataPrsnt($'Add') == false)
    {
        // wait until the data "Add" becomes available
    }

    Start("://Sequence Control/Sequence Control 04.asf.xts", "DoNotWait");

    // wait until both calculations have finished
    WaitForCF("://Sequence Control/Sequence Control 02.amf.xts", "Stopped";);
    WaitForCF("://Sequence Control/Sequence Control 04.asf.xts", "Stopped";);

    // stop the Device Profiles
    Stop("://ION VIB-A T001.dpf.xts");
    Stop("://IFN VIB-ACC.dpf.xts");

    // stop the Interface Profile
    Stop("://Simulation T001.ipf.xts");

    // remove the calculated data
    RemoveData("://Sequence Control/Sequence Control 02.amf.xts");
    RemoveData("://Sequence Control/Sequence Control 04.asf.xts");
}
```

The following listing contains the main loop of the example Analyzing Script “Sequence Control 01”:

```
var main()
{
    while(1)
    {
        // run our three different measurement and analyzing tasks forever
        PerformSequentialCalculations();
        PerformParallelCalculations();
        PerformDependentCalculations();
    }
}
```

Try it out

- ➔ Start the Analyzing Script “Sequence Control 01” from the example location (“Files\Example Location\Analyzing Script Files\Sequence Control 01”).
- ➔ See how the different Configuration Files are being started and stopped.
- ➔ Check the offline data which is being stored by each execution of the measurement and analyzing tasks.
- ➔ Add your own functionality to the participating Configuration Files in order try out additional possibilities.

Ready

At this point you know how to use the basic mechanisms of the sequence control. Refer to the descriptions of the available Analyzing Functions and to the best practices in order to learn more about the detailed functionalities and about the most common use cases.

6.21 Opening of Monitoring Views via the Command Line

6.21.1 General Description

In this chapter you will learn how to open the ***X-Tools Client*** with an additional command line parameter so that it automatically opens the specified Monitoring View.

This chapter contains the following topics:

- **Command Line Syntax**

6.21.2 Command Line Syntax

Prepare a Monitoring View

- Ensure that you have a Monitoring View with the name “Open me” below the “Default Location” of the ***X-Tools Server***.
- Close the ***X-Tools Client*** and the ***X-Tools Server***.

Functionality

- In order to open a Monitoring View automatically after the startup of the ***X-Tools Client***, the “/openMVF” command line parameter is provided.
- The syntax of “/openMVF” is defined as follows:
  - /openMVF “*ServerName* - *LocationName*://[*Path*/]*FileName* [/fullscreen]
    - The /fullscreen command line parameter is optional and can be used in order to open the specified Monitoring View in full screen mode.
- Examples for the command line parameters:
  - /openMVF “YourServer - Default Location://Open me”
  - /openMVF “YourServer - Default Location://SubDirectory/Open me” /fullscreen
- In case multiple Monitoring Views shall be opened, multiple “/openMVF” command line parameters with to-be-opened Monitoring Views can be specified.
- Examples for the full command line:
  - “C:\CMS X-Tools Pro\Client\Bin\X-Tools Client Pro.exe” /openMVF “YourServer - Default Location://Open me”
  - “C:\CMS X-Tools Pro\Client\Bin\X-Tools Client Pro.exe” /openMVF “YourServer - Default Location://Open me” /openMVF “YourServer - Default Location://SubDirectory/Open me”
- In case the ***X-Tools Client*** is running already when such a command line is ran again, the already running ***X-Tools Client*** is re-used and displays the specified Monitoring Views.
  - All Monitoring Views which are not part of the command line are closed automatically, no matter whether they contain unsaved changes at the moment or not.

Try it out

➔ Close the ***X-Tools Client*** and the ***X-Tools Server***.

➔ Create a shortcut to the executable of the ***X-Tools Client*** (“X-Tools Client Pro.exe”).

➔ Edit the shortcut and add the “/openMVF” command line parameters together with the path to an existing Monitoring View.

➔ Save and execute the shortcut.

The ***X-Tools Client*** is being started and starts the ***X-Tools Server*** automatically. In addition, the specified Monitoring View File is being opened and visualizes the previously configured data.

Ready

At this point you know how to use the command line parameters of the ***X-Tools Client*** in order to automatically open a Monitoring View after the startup of the ***X-Tools Client***.

6.22 Printing of Monitoring Views via the Command Line

6.22.1 General Description

In this chapter you will learn how to open the ***X-Tools Client*** with an additional command line parameter so that it automatically prints the specified Monitoring View.

This chapter contains the following topics:

- **Command Line Syntax**

6.22.2 Command Line Syntax

Prepare a Monitoring View

- Ensure that you have a Monitoring View with the name “Print me” below the “Default Location” of the ***X-Tools Server***.
- Close the ***X-Tools Client*** and the ***X-Tools Server***.

Functionality

- In order to print a Monitoring View automatically after the startup of the ***X-Tools Client***, the “/printMVF” command line parameter is provided.
- The syntax of “/printMVF” is defined as follows:
  - /printMVF “*ServerName* - *LocationName*://[*Path*/]*FileName*
- Examples for the command line parameters:
  - /printMVF “YourServer - Default Location://Print me”
  - /printMVF “YourServer - Default Location://SubDirectory/Print me”
- In case multiple Monitoring Views shall be printed, multiple “/printMVF” command line parameters with to-be-printed Monitoring Views can be specified.
- Examples for the full command line:
  - “C:\CMS X-Tools Pro\Client\Bin\X-Tools Client Pro.exe” /printMVF “YourServer - Default Location://Print me”
  - “C:\CMS X-Tools Pro\X-Tools\Client\Bin\X-Tools Client Pro.exe” /printMVF “YourServer - Default Location://Print me” /printMVF “YourServer - Default Location://SubDirectory/Print me”
- In case the ***X-Tools Client*** is running already when such a command line is ran again, the already running ***X-Tools Client*** is re-used and prints the specified Monitoring Views.
  - The actual printing is performed through the ***X-Tools Server***, thus the printing also can be performed at a remote computer in case the ***X-Tools Server*** is running at this remote computer.
  - Typically, the ***X-Tools Server*** should be running already when this command line parameter is used, because it would not be able to show a lot of data values at the printout in case it would have been started only a few moments ago.
- The printing is performed via the default printer which is currently configured for the computer and Windows user at which the ***X-Tools Server*** is running.
  - When a different default printer shall be configured and used by the ***X-Tools Server***, the default printer must be changed for the computer and user at which the ***X-Tools Server*** is running (and not for the computer and user at which the ***X-Tools Client*** is running).

Try it out

➔ Ensure that ***X-Tools Client*** and ***X-Tools Server*** are running.

➔ Create a shortcut to the executable of the ***X-Tools Client*** (“X-Tools Client Pro.exe”).

➔ Edit the shortcut and add the “/printMVF” command line parameters together with the path to an existing Monitoring View.

➔ Save and execute the shortcut.

The ***X-Tools Server*** prints the specified Monitoring View File.

Ready

At this point you know how to use the command line parameters of the ***X-Tools Client*** in order to automatically print a Monitoring View via the ***X-Tools Server***.

6.23 Storing of Data Snapshots out of the Monitoring System

6.23.1 General Description

In this chapter you will learn how to store a snapshot of the data which is currently being visualized by the **Monitoring System**.

This chapter contains the following topics:

- **Storing of Data Snapshots**

6.23.2 Storing of Data Snapshots

Open a Monitoring View

- Open a Monitoring View and display data with at least two different **Monitoring Charts**, for example one **MTC yt T001** and one **MTC yx T001**.

Functionality

- Press the Store Data Snapshot button from the Toolbar Area of one **Monitoring Chart** in order to create a snapshot of the currently visualized data.
- The functionality of the data snapshot can be configured to either store only the data from the **Monitoring Chart** whose Store Data Shapshot button has been pressed or to store all of the data from all **Monitoring Charts** of the current Monitoring View.
  - The scope of the data snapshot (current **Monitoring Chart** or current Monitoring View) can be configured via the Data Snapshot Scope menu item of the context menu of the Store Data Snapshot button and via the Data Snapshot Scope parameter of the **Chart Options** dialog.
- In case of **Monitoring Charts** which display a time interval of data (e.g. the **MTC yt T001** or the **MTC yx T001**), the currently displayed time interval of the currently displayed data is being stored.
- In case of **Monitoring Charts** which display a point in time (e.g. the **MTC yn T001** or the **MTC ynm T001**), the currently displayed point in time of the currently displayed data is being stored.
- After the Store Data Snapshot button has been pressed, a copy of the to-be-stored data is being created automatically. After the storing has been completed, the memory which has been used for the copy is freed again.
- While the storing is going on, the **Storage Progress** dialog is being shown and displays information about the progress of the storing. The Cancel button of the **Storage Progress** dialog can be used in order to cancel the ongoing storing.
- Multiple data snapshots can be stored in parallel in case the Store Data Snapshot button is pressed multiple times.
- The data snapshot can be created for all types of data, for offline data and for paused or running online data.
- All offline data files which are created by a data snapshot are stored into one common directory. The name of the created directory starts with “Data Snapshot” and also contains the date and time of the storing.
- The name of each offline data file consists out of the name of the data and of the date and time of the moment of the storing in order to receive unique file names also for scenarios where the same data name is present within the issuing **Monitoring Chart** multiple times.
- All offline data files are being stored to the offline data location which was used by the last store or advanced store operation.

Try it out

- ➔ Display some data within a **Monitoring Chart**.
- ➔ Store a snapshot of the displayed data.
- ➔ After the snapshot has been stored, load the stored offline data.
- ➔ Display the loaded data within a new **Monitoring Chart** and verify that the correct values and times of all data have been stored.

Ready

At this point you know how to use the data snapshot functionality in order to store all of the values of the data which are currently visualized by a **Monitoring Chart**.

6.24 Receiving of Data from standard OPC UA Servers

6.24.1 General Description

In this chapter you will learn how to receive data from standard OPC UA servers.

This chapter contains the following topics:

- **Connecting to an OPC UA Server**
- **Configuring of the Data of interest**
- **Using of the Data of interest**

6.24.2 Connecting to an OPC UA Server

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Functionality

- Within the **DMS Explorer**, expand the “Main -> Servers -> *YourServer* -> Interfaces” branch.
- Click onto the present “OPC UA T001” item with the right mouse button in order to open the context menu.
- Choose New Interface Profile from the context menu.
  - An **IPE OPC UA T001** with a new, empty Interface Profile opens automatically.
- Within the Connections table, enter the name or IP of the target OPC UA server into the Target IP or Name field and its port into the Port field.
- Configure the values for Server Certificate, Security Policy, Message Security, User Name and Password to the values which are required for the successful acces to your OPC UA server.
- In case there is no OPC UA client certificate present at the ***X-Tools Server*** yet, use the OPC UA Client Certificate table in order to create a new OPC UA client certificate - simply call the context menu from the title bar of the OPC UA Client Certificate table or click into any of the present Value fields.
- Save the new Interface Profile.
- Start the new Interface Profile.
  - The configured OPC UA server appears within the **DMS Explorer** below the “Main -> Servers -> *YourServer* -> Interfaces -> OPC UA T001” branch.
- For a more detailed description of all available functionalities of the **IPE OPC UA T001**, refer to the reference manual of the “Device Management System”, chapter “IPE OPC UA T001”.

Try it out

➔ Create a new Interface Profile in order to connect to an existing OPC UA server.

➔ Start the new Interface Profile.

The configured OPC UA server becomes present within the **DMS Explorer**.

Ready

At this point you know how to connect to standard OPC UA servers.

6.24.3 Configuring of the Data of interest

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Functionality

- Within the **DMS Explorer**, expand the “Main -> Servers -> *YourServer* -> Interfaces -> OPC UA T001” branch.
- Click onto the connected OPC UA server with the right mouse button in order to open the context menu.
- Choose New Device Profile from the context menu.
  - A **DPE OPC UA T001** with a new, empty Device Profile opens automatically.
- Within the Address Space table, the current address space of the OPC UA server can be browsed. The Add as Device Profile Input Data and Add as Device Profile Output Data context menu items can be used in order to configure any of the data of the OPC UA server to be used as input and/or output data.
  - The values of all data which are added to the Device Profile Input Data table become available as online data within the **MDS Explorer** after the Device Profile has been started.
  - The values of all data which are added to the Device Profile Output Data table are being written cyclicly to the OPC UA server after the Device Profile has been started. In order to define which values are to be written, the online data which shall act as source for the to-be-sent values must be entered into the Assigned Data field.
- The Value field of the Device Profile Output Data table can be used in order to change the value of the according data of the OPC UA server instantly, no matter whether any Device Profile is currently running or not.
- Save the new Device Profile.
- Start the new Device Profile.
  - All of the data which is present and enabled within the Device Profile Input Data table becomes present within the **MDS Explorer** as online data.
- For a more detailed description of all available functionalities of the **DPE OPC UA T001**, refer to the reference manual of the “Device Management System”, chapter “DPE OPC UA T001”.

Try it out

➔ Create a new Device Profile in order to receive the data from the existing OPC UA server.

➔ Start the new Device Profile.

The configured online data becomes present within the **MDS Explorer**.

Ready

At this point you know how to receive online data from standard OPC UA servers.

6.24.4 Using of the Data of interest

Functionality

- The online data which is received from an OPC UA server can be visualized like any other online data within the system.
- The online data also can be used within the **Analyzing System** and the **Storage System** of ***X-Tools*** like any other online data.

Try it out

➔ Visualize the data from the OPC UA server within the **Monitoring System**.

➔ Perform simple calculations on base of the online data from the OPC UA server.

➔ Store some of the online data from the OPC UA server.

Ready

At this point you know how to use the online data from standard OPC UA servers.

6.25 Providing of Data to standard OPC UA Clients

6.25.1 General Description

In this chapter you will learn how to use the built-in OPC UA server of **X-Tools** in order to provide all of the current online data to standard OPC UA clients.

This chapter contains the following topics:

- **Enabling of the built-in OPC UA Server**
- **Using of the built-in OPC UA Server**

6.25.2 Enabling of the built-in OPC UA Server

Open the MMS Explorer

- Open the **Main Management System** and ensure that the **MMS Explorer** is displayed at the left side of the **Main Management System** workspace.

Functionality

- Within the **Main Explorer**, expand the “Main -> Servers” branch.
- Click onto the item which represents your server with the right mouse button in order to open the context menu.
- Choose Edit Global Options from the context menu.
  - A **Global Options Editor** with the current settings of your server opens automatically.
- Configure the setting Enable OPC UA Server to “Yes” in order to enable the built-in OPC UA server.
- Configure the setting Port to the port which shall be used by the built-in OPC UA server.
- Press the Save button from the menu in order to take over your changes.
  - Changes of the used OPC UA port are taken over only at the start of the OPC UA server. In order to change the port of an already running OPC UA server, disable the OPC UA server first and change the to-be-used port and re-enable the OPC UA server then.

Note

Internally, the OPC UA server connects to the rest of the ***X-Tools Server*** like a standard client. Therefore, each enabling and disabling of the built-in OPC UA server creates log entries about the attach and detach of a client and also increases or decreases the number of connected clients which is visible within the **MMS Explorer**.

Note

The OPC UA server uses the user “Anonymous” in order to connect to the rest of the ***X-Tools Server***. The only required access right is “Read Access” for the “Online Data” functionality. Therefore, all other access rights can be removed for the user “Anonymous” in case the ***X-Tools Server*** must be protected against unauthorized access.

Try it out

➔ Enable the built-in OPC UA server.

Ready

At this point you know how to enable and to disable the built-in OPC UA server.

6.25.3 Using of the built-in OPC UA Server

Functionality

- After the built-in OPC UA server has been enabled it can be accessed with standard OPC UA clients.
- In order to connect to the built-in OPC UA server with a standard OPC UA client, enter the IP of your PC and the configured OPC UA port within the according mask of your OPC UA client.

Try it out

➔ Use a standard OPC UA client in order to connect to the built-in OPC UA server.

➔ Read values from different online data of ***X-Tools*** via the connected OPC UA client.

Ready

At this point you know how to provide the online data of ***X-Tools*** to standard OPC UA clients.

6.26 Creation of Screenshots out of the Monitoring System

6.26.1 General Description

In this chapter you will learn how to print and save a screenshot of the data which is currently being visualized by the **Monitoring System**.

This chapter contains the following topics:

- **Printing of Screenshots**
- **Saving of Screenshots**

6.26.2 Printing of Screenshots

Open a Monitoring View

- Open a Monitoring View and display data with at least two different **Monitoring Charts**, for example one **MTC yt T001** and one **MTC yx T001**.

Functionality

- Call the context menu from the Curve Area of any present Monitoring Chart and choose Print Screenshot.
- The standard **Print** dialog of Windows opens and allows to choose which printer and which settings shall be used for the printout.

Try it out

➔ Display some data within a **Monitoring Chart**.

➔ Print a screenshot of the displayed data.

➔ After the screenshot has been printed, verify that the printout matches the original Monitoring View.

Ready

At this point you know how to use the screenshot functionality in order to print the appearing of the currently visualized Monitoring View.

6.26.3 Saving of Screenshots

Open a Monitoring View

- Open a Monitoring View and display data with at least two different **Monitoring Charts**, for example one **MTC yt T001** and one **MTC yx T001**.

Functionality

- Call the context menu from the Curve Area of any present Monitoring Chart and choose Save Screenshot.
- The screenshot of the current Monitoring View is being saved to the "..\X-Tools\Client\User\Screenshot" directory of your ***X-Tools*** installation.

Try it out

➔ Display some data within a **Monitoring Chart**.

➔ Save a screenshot of the displayed data.

➔ After the screenshot has been saved, open the created file and verify that it matches the original Monitoring View.

Ready

At this point you know how to use the screenshot functionality in order to save the appearing of the currently visualized Monitoring View.

6.27 Creation of Automatic Reports out of the Monitoring System

6.27.1 General Description

In this chapter you will learn how to create images of the current visualizations within Monitoring Views automatically, based on a trigger data and on additional trigger information which is provided to the Monitoring View.

Whenever a Monitoring View with offline data as input runs into an error, it stops automatically. This is different to Monitoring View with online data as input, which suspend in such a situation and attempt to resume after the error condition is gone. The main reasons for this different kind of handling are:

- for Monitoring Views with offline data as input it is much less likely that the error condition will disappear ever
- a main use cases for Monitoring Views with offline data as input are sequential Analyzing Scripts (see point 6.20), which could be blocked by a suspended Monitoring View without any benefit

This chapter contains the following topics:

- **Important Facts**
- **Creation of a Monitoring View for automatic Reports**
- **Trigger Examples**
- **Example Configuration Files**

6.27.2 Important Facts

After the functionality of automatic reports has detected another event, it waits the configured capture delay before the report is being created. Thus, the report is taken from the time "ReportTime" = "TimeOfTheTriggerEvent" + "CaptureDelay", the visible time interval within the report is from "ReportTime" back into the past.

The report always covers the time interval which is currently configured within the Monitoring View (not "TimeInterval" + "CaptureDelay"). There is no other possibility to predefine the time interval which shall be captured by the automatic report, thus in theory it also can be configured that the time of the actual event is not present within the report anymore.

Automatic reports through Monitoring Views are a pure thing of the ***X-Tools Server***. The ***X-Tools Client*** does not take part in the creation or handling of this kind of reports at all.

Each report is stored to the location which has been configured within the Automatic Reports Parameters table. The path below the specified location contains the following parts:

- absolute path of the chosen Storage Location +
- chosen Storage Path

The file name of each report contains the following parts:

- chosen Storage File Name +
- timestamp at which the report has been created
  - The report is being created at the moment of the trigger event + the capture delay. However, the date information within the file name points to the moment of the trigger event.

Example 1:

- Location which is specified within the Automatic Reports Parameters table: “Example Location”
- absolute path of the location “Example Location”: “C:\Example Location”
- storage path of the Monitoring View: “Example Location://Monitoring View Files\My Directory\Subfolder”
- Storage Path = “Default”
- Storage File Name = “Default”
- Example file name (full path):
  - C:\Example Location\MTE\Reports\My Directory\Subfolder\My Monitoring View 2011-01-28 12-53-00.000.000.000.png

Example 2:

- Location which is specified within the Automatic Reports Parameters table: “Example Location”
- absolute path of the location “Example Location”: “C:\Example Location”
- storage path of the Monitoring View: “Example Location://Monitoring View Files\My Directory\Subfolder”
- Storage Path = “MTE\Reports\Collector”
- Storage File Name = “Temp”
- Example file name (full path):
  - C:\Example Location\MTE\Reports\Collector\Temp 2011-01-28 12-53-00.000.000.000.png

The following rules apply for the created reports:

- The width of the Monitoring View, in number of pixels, matches the value of the Width cell from the Automatic Reports Parameters table.
- The height of the Monitoring View, in number of pixels, matches the value of the Height cell from the Automatic Reports Parameters table.
- The file format and the file extension of the created file is .png.
- A report file does not contain the measurement cursors of any Monitoring Chart, but in case the advanced cursors and/or frequency markers of a **MTC yn T001** were turned on at the moment when the Monitoring View was saved, they are added also to the report file.

6.27.3 Creation of a Monitoring View for automatic Reports

Open a Monitoring View

- Open a Monitoring View and display data with at least two different **Monitoring Charts**, for example one **MTC yt T001** and one **MTC yx T001**.

Functionality

- Within the Automatic Reports Parameters table, configure the desired width, height and target location for the created automatic report files.
- Within the Automatic Reports Trigger table, specify the name of the to-be-used trigger data and the conditions at which the automatic reports shall be created.

- The Monitoring View monitors the configured trigger event and creates reports automatically after it has been started.
- Whenever the configured trigger condition becomes true, the Monitoring View stores another report file to the specified target location.

Try it out

➔ Display some data within a **Monitoring Chart**.

➔ Configure a trigger data and a trigger condition which is fulfilled from time to time.

➔ Save the Monitoring View and start it.

➔ After a report has been created, verify that the saved report matches the status of the system at the moment at which the trigger condition became true.

➔ Set the value of Define available Width manually within the Monitoring View Parameters table to “Yes (use the Width from the Automatic Reports Parameters Table)” and become able to edit the present Monitoring Charts with the exact width of the to-be-created reports.

➔ Set the value of Define available Height manually within the Monitoring View Parameters table to “Yes (use the Height from the Automatic Reports Parameters Table)” and become able to edit the present Monitoring Charts with the exact height of the to-be-created reports.

Ready

At this point you know how to configure Monitoring Views for the creation of automatic reports.

6.27.4 Trigger Examples

The following examples show how the Automatic Reports Trigger table can be configured in order to cover different scenarios.

- Scenario 1:
  - The report shall be created
    - … only once
    - … as soon as the value of the online data “Report Trigger” is above 100
    - … with a delay of 5 seconds
  - The configuration within the Automatic Reports Trigger table can be as follows:

Parameter	Value
Trigger Mode	Once
Time Domain	Online Data with absolute Timestamps
Trigger Data Name	Report Trigger
Trigger Type	Threshold
Threshold Type	Value > Threshold
Threshold Value	100
Capture Delay Format	s
Capture Delay Value	5

- Scenario 2:
  - The report shall be created
    - … repeating
    - … as soon as the value of the online data “Report Trigger” is between 50 and 100
    - … with a hysteresis of 10
    - … with a delay of 5 seconds
  - The configuration within the Automatic Reports Trigger table can be as follows:

Parameter	Value
Trigger Mode	Repeating
Time Domain	Online Data with absolute Timestamps
Trigger Data Name	Report Trigger
Trigger Type	Interval
Interval Type	Inside Interval
Lower Limit	50
Upper Limit	100
Capture Delay Format	s
Capture Delay Value	5

- Scenario 3:
  - The report shall be created
    - … repeating
    - … as soon as the binary online data “Report Trigger” shows a rising edge
    - … with a delay of 5 seconds
  - The configuration within the Automatic Reports Trigger table can be as follows:

Parameter	Value
Trigger Mode	Repeating
Time Domain	Online Data with absolute Timestamps
Trigger Data Name	Report Trigger
Trigger Type	Threshold
Threshold Type	Rising Edge with Threshold
Threshold Value	0.5
Hysteresis Value	0.1
Capture Delay Format	s
Capture Delay Value	5

6.27.5 Example Configuration Files

The Monitoring View “Example for an Automatic Report” from the example location provides an example for automatic reports. In order to receive a report via this Monitoring View, the following Configuration Files from the example location must be running:

- the Interface Profile “Simulation T001”
- the Device Profile “ION AnalogInput T001”
- the Analyzing Model “Abs 01”
- the Analyzing Model “AddV2D 01”
- the Analyzing Model “H2D 01”
- the Analyzing Model “FFT 01”

The report which is created through this Monitoring View looks like in the following example:

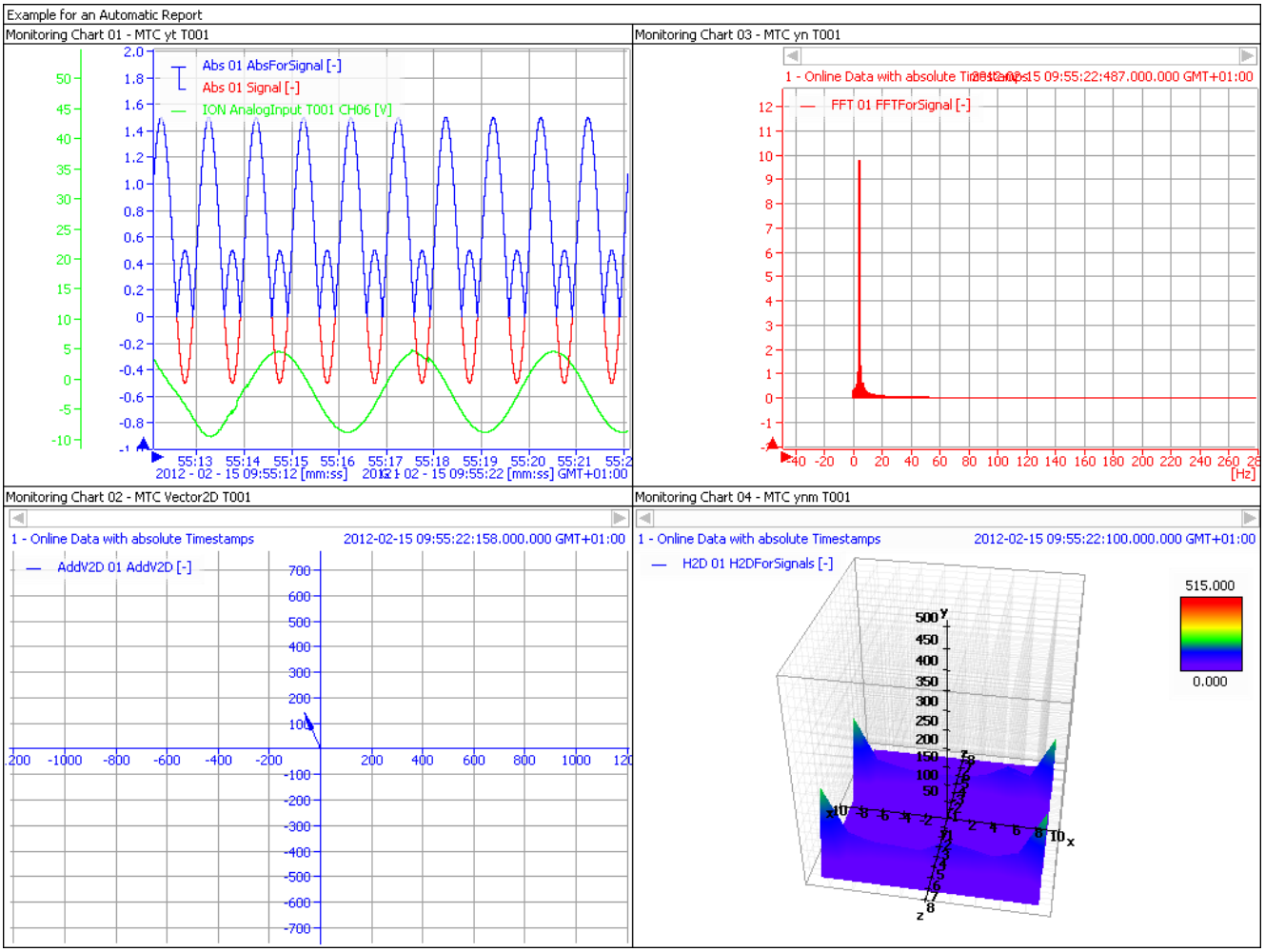

Example for an Automatic Report
Monitoring Chart 01 - MTC yt T001
Abs 01 AbsForSignal [-]
Abs 01 Signal [-]
ION AnalogInput T001 CH06 [V]
Monitoring Chart 03 - MTC yn T001
FFT 01 FFTForSignal [-]
[Hz]
Monitoring Chart 02 - MTC Vector2D T001
1 - Online Data with absolute Timestamps
2012-02-15 09:55:22:158.000.000 GMT+01:00
AddV2D 01 AddV2D [-]
Monitoring Chart 04 - MTC ynm T001
1 - Online Data with absolute Timestamps
2012-02-15 09:55:22:100.000.000 GMT+01:00
H2D 01 H2DForSignals [-]
515.000
0.000

Figure 59: Example for an Automatic Report

6.28 Creation of Reports of Analyzing Models

6.28.1 General Description

In this chapter you will learn how to print and save reports of Analyzing Models via the **AME Standard T001**.

This chapter contains the following topics:

- **Important Facts**
- **Printing of a Report**
- **Saving of a Report**

6.28.2 Important Facts

Reports of Analyzing Models are a pure thing of the ***X-Tools Client***. The ***X-Tools Server*** does not take part in the creation or handling of this kind of reports at all.

Each report consists out of a draft which shows the complete Analyzing Model at a zoom level of 100 % and of a HTML document which lists all of the information about the Analyzing Model and all of its contained (Analyzing and User) Functions.

Each saved report is stored to the following location:

- ...\X-Tools\Client\User\Reports

The file names of each saved report contain the following parts:

- name of the Analyzing Model
- timestamp at which the report has been created

Example:

- My Analyzing Model 2011-12-05 10-07-00.000.000.000.png
- My Analyzing Model 2011-12-05 10-07-00.000.000.000.htm

6.28.3 Printing of a Report

Open an Analyzing Model

- Open an Analyzing Model which contains multiple (Analyzing and/or User) Functions.

Functionality

- Reports can be sent to any of the currently available printers via the Print Report context menu item within the Action Area of the **AME Standard T001**.

Try it out

➔ Put multiple (Analyzing and/or User) Functions into an AME Standard T001.

➔ Print a report of the current Analyzing Model.

The standard printer dialog of Windows pops up and allows to choose the printer. The printed document contains all information about the Analyzing Model and its contained (Analyzing and/or User) Functions as well as a draft of the complete Analyzing Model.

Ready

At this point you know how to print reports of Analyzing Models.

6.28.4 Saving of a Report

Open an Analyzing Model

- Open an Analyzing Model which contains multiple (Analyzing and/or User) Functions.

Functionality

- Reports can be saved to file via the Save Report context menu item within the Action Area of the **AME Standard T001**.

Try it out

➔ Put multiple (Analyzing and/or User) Functions into an AME Standard T001.

➔ Save a report of the current Analyzing Model.

➔ Open the created HTML document.

The HTML document contains all information about the Analyzing Model and its contained (Analyzing and/or User) Functions as well as a draft of the complete Analyzing Model.

Ready

At this point you know how to save reports of Analyzing Models.

6.29 Fast Analysis via the MDS Explorer

6.29.1 General Description

In this chapter you will learn how to prepare the system in order to become able to start fast analysis tasks for online and offline data via the **MDS Explorer**.

This chapter contains the following topics:

- **Creation of a Location for Fast Analysis Scripts**
- **Creation of a new Fast Analysis Script**
- **Opening of existing Fast Analysis Scripts**
- **Using of Fast Analysis Scripts**

6.29.2 Creation of a Location for Fast Analysis Scripts

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.

Functionality

- Expand the “Main -> Servers -> *YourServer*” branch of the **ANS Explorer**.
- In order to add a Fast Analysis location, call the context menu for the “Files“ item within the **ANS Explorer** and choose Add Location….
- Within the upcoming **Add Location** dialog, choose any symbolic name for your new Fast Analysis location and specify the path to this location at the disk. In addition, change the value of Location Type to “Fast Analysis”.
- Press OK in order to add the specified location to the ***X-Tools Server***.
- After another Fast Analysis location has been added successfully, the ***X-Tools Server*** automatically creates all of the necessary sub-directories.
  - Locations of type “Fast Analysis” do not support any Configuration Files except Analyzing Scripts. Therefore, locations of this type are visible only within the **ANS Explorer** and do not contain any sub-items except “Analyzing Script Files”.
  - Only one location of type “Fast Analysis” can be added to the ***X-Tools Server*** at any time.

Try it out

➔ Add the “..\CMS X-Tools Pro\Server\User\Fast Analysis Location” directory as Fast Analysis location.

All of the Analyzing Scripts from the location become visible within the **ANS Explorer** of the ***X-Tools Server***.

Ready

At this point you know how to add a Fast Analysis location to the ***X-Tools Server***.

6.29.3 Creation of a new Fast Analysis Script

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Storage System** workspace.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” and “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branches of the **MDS Explorer**.

Functionality

- Fast Analysis Scripts are created and maintained in exactly the same way as normal Analyzing Scripts (see point 6.14). The main difference for the creation and maintaining is that normal Analyzing Scripts are stored to locations of type “Configuration Files” and Fast Analysis Scripts are stored to locations of type “Fast Analysis”.
- The following things are special for Fast Analysis Scripts:
  - Fast Analysis Scripts do not need pre-defined input data. Instead, variables can be used for input data - when a Fast Analysis is started via the **MDS Explorer**, these variables are filled with the currently chosen (online or offline) data and the calculation is performed on base of this data.
  - The syntax for variables which shall represent input data for the Fast Analysis Script is _’INPUT_DATA_x’, where x = 1 for the first selected data from the MDS Explorer, x = 2 for the second selected data, x = 3 for the third selected data and so on.
    - The syntax like _’INPUT_DATA_1’ can be used everwhere where also the direct reference to data (like ~’Online Data’ or $’Offline Data’) can be used.
  - There is no way to define the names of the output data of Fast Analysis Scripts dynamically. Thus, the names of all output data of a Fast Analysis Script are defined at the moment when the Fast Analysis Script is created.
  - When the context menu of the **MDS Explorer** is called from above an online data, it provides only those Fast Analysis Scripts whose Output Data Class is “Online Data”. Accordingly, the context menu provides only those Fast Analysis Scripts whose Output Data Class is “Offline Data” when it is called from above an offline data.

Try it out

➔ Create a new Fast Analysis Script.

➔ Build an easy Analyzing Script which takes two variable input data, adds them together and outputs the result.

➔ Save the Analyzing Script to the “Fast Analysis” location. At the first save, the **Save As** dialog will pop up and offers a default file name. Change the default file name to “Hello Fast Analysis” and press OK in order to save it to the disk.

The new Fast Analysis Script now is displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Script Files” branch of the **ANS Explorer**.

➔ Take a look onto the Analyzing Scripts which are provided via the “Fast Analysis Location” in order to see application examples about the capabilities and usage of Fast Analysis Scripts.

Ready

At this point you know how to create a new Fast Analysis Script which can be used in order to perform calculations based on online and/or offline data.

6.29.4 Opening of existing Fast Analysis Scripts

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.

Functionality

- Within the **ANS Explorer**, expand the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Script Files” branch. *YourLocation* must be a location of type “Fast Analysis”.
- Drag one of the below Fast Analysis Scripts with the left mouse into the empty space at the right side of the **ANS Explorer**. An **Analyzing Script Editor** of proper type is opened automatically and displays the dropped Fast Analysis Script.

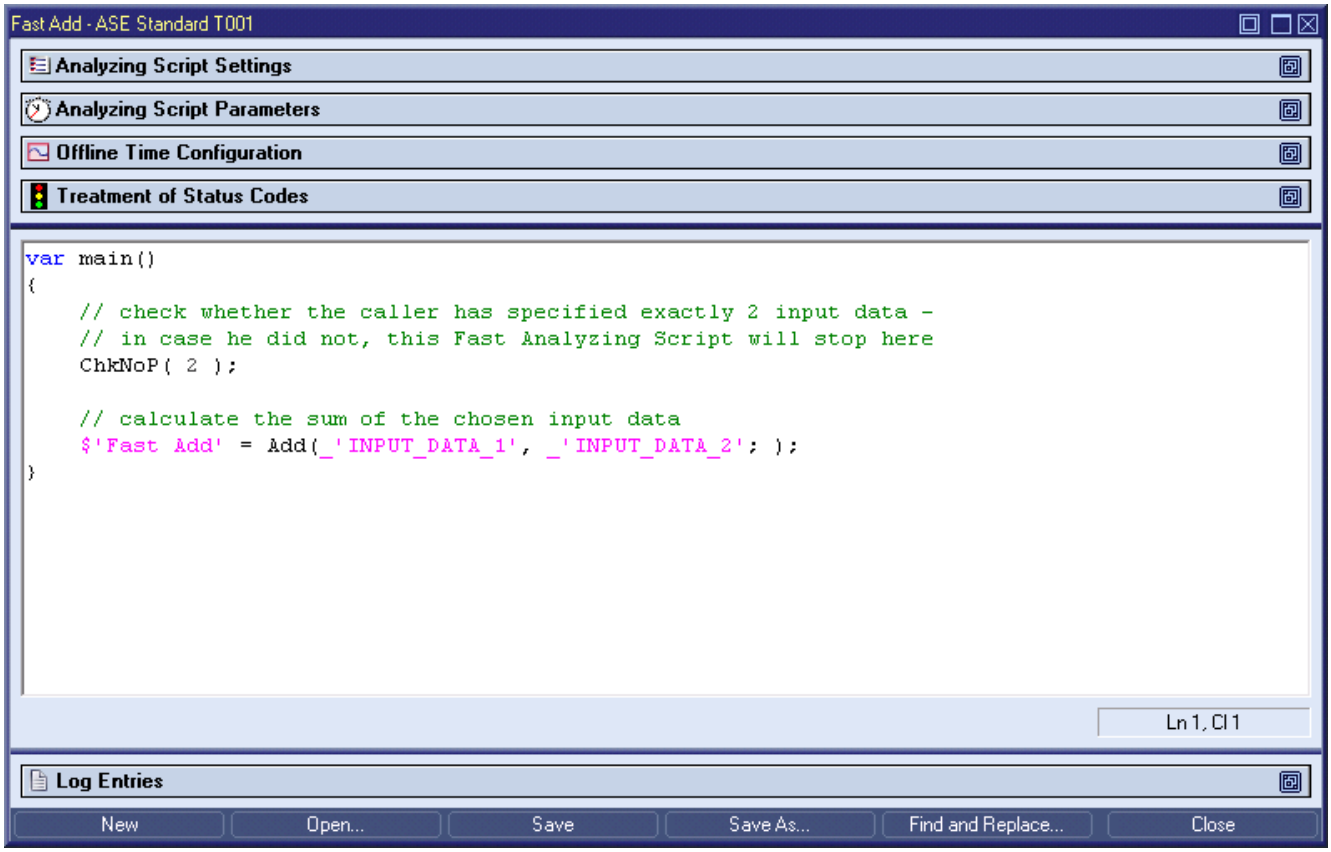


*Figure 60: **ASE Standard T001** with an opened Fast Analysis Script*

- The opened Fast Analysis Script can be edited now. After editing, the Save and Save As… buttons are used in order to save the edited Fast Analysis Script with all of its settings.

Try it out

➔ Open an existing Fast Analysis Script.

➔ Enter another storage location and change the Script Description within the Analyzing Script Settings table.

➔ Add an additional function to the Fast Analysis Script and output also this additional result of the calculation.

➔ Save the Fast Analysis Script via the Save button.

The modified Fast Analysis Script is still displayed below the “Main -> Servers -> *YourServer* -> *YourLocation* -> Analyzing Script Files” branch of the **ANS Explorer**.

Ready

At this point you know how to open and modify an existing Fast Analysis Script.

6.29.5 Using of Fast Analysis Scripts

Open the MDS Explorer

- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Online Data -> All Online Data” and “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branches of the **MDS Explorer**.
- Ensure that you have some Fast Analysis Scripts with an Output Data Class of “Online Data” and some Fast Analyzing Scripts with an Output Data Class of “Offline Data” available within a “Fast Analysis” location.

Functionality

- Within the **MDS Explorer**, call the context menu from an online data and choose any of the sub-items of the Fast Analysis context menu item.
- The **Fast Analysis Status** dialog opens automatically and displays the current status of the ongoing fast analysis.
  - The Show Log button can be used in order to open a **System Log Viewer** which contains all log entries that are dependent to the running analysis.
  - The Cancel button can be used in order to stop the running analysis and to close the **Fast Analysis Status** dialog.
- Within the **MDS Explorer**, call the context menu from an offline data and choose any of the sub-items of the Fast Analysis context menu item.
- The **Fast Analysis Status** dialog opens automatically and displays the current status of the ongoing fast analysis.
  - The Show Log button can be used in order to open a **System Log Viewer** which contains all log entries that are dependent to the running analysis.
  - The Cancel button can be used in order to stop the running analysis and to close the **Fast Analysis Status** dialog.
- The following things must be considered for the usage of Fast Analysis Scripts:
  - Running of a Fast Analysis Script is a temporary operation - therefore, Fast Analysis Scripts are not remembered for to-be-restarted during the next startup of the ***X-Tools Server***, even when they were running at the moment of the shutdown of the ***X-Tools Server***.
  - It is not possible to use the same Fast Analysing Script for more than one analysis at a time.
  - The **Fast Analysis Status** dialog is shown only by the ***X-Tools Client*** which has been used for the start of the analysis.
  - Fast Analysis Script can be stopped and restarted via the **ANS Explorer**.
  - The current status of each Fast Analysis Script (running or stopped) is visible within the **ANS Explorer**.
  - In case a Fast Analysis Script would have to suspend, the fast analysis is stopped with an error (it won't resume).
  - In case any fast analysis finishes with success, the **Fast Analysis Status** dialog is closed automatically. Contrary, the **Fast Analysis Status** dialog remains open in case a fast analysis ends with an error.

Try it out

➔ Start a fast analysis for online data.

The output data which has been configured within the used Fast Analysis Script becomes available within the **MDS Explorer** and the **Fast Analysis Status** dialog is opened.

➔ View the log entries which belong to the currently running analysis.

➔ Cancel the currently running analysis.

The data which has been configured within the used Fast Analysis Script is being removed from the **MDS Explorer** and the **Fast Analysis Status** dialog is closed.

➔ Start a fast analysis for offline data.

The output data which has been configured within the used Fast Analysis Script becomes available within the **MDS Explorer** and the **Fast Analysis Status** dialog is opened. In case the offline calculation is perfomed only once, the **Fast Analysis Status** dialog disappears automatically after the offline calculation has finished.

Ready

At this point you know how to use existing Fast Analysis Scripts in order to start and stop the calculations based on the currently available online and/or offline data.

6.30 Calculation of Formulas via the MVE Standard T001

6.30.1 General Description

In this chapter you will learn how to use the Formula Editor within the **MVE Standard T001** in order to prepare and run simple calculations directly within the **Monitoring System**.

This chapter contains the following topics:

- **Creation of a simple Formula**
- **Running of Formulas**

6.30.2 Creation of a simple Formula

Open the MTS Explorer

- Ensure that there are a few online and offline data of type Boolean, Integer and/or decimal available within the **MDS Explorer**.
- Open the **Monitoring System** and ensure that the **MTS Explorer** is displayed at the left side of the **Monitoring System** workspace.
- Drag&Drop an online data into the empty **Monitoring System** workspace in order to open a new **MVE Standard T001**.

Functionality

- Expand the Formula Editor table.
- Enter a new name into the Name cell, e.g. “My first Formula”.
  - After the name has been entered, the Formula cell becomes enabled and another, empty row is added to the Formula Editor table.
- Drag&Drop the online data which has been used in order to open the **MVE Standard T001** into the Formula cell.
- Add something to the data, e.g. “ + 100”. After this step, the Formula cell contains something like this:
  - ~'Some online Data' + 50
- Enter another name into the Name cell of the second row, e.g. “My second Formula”.
  - After the name has been entered, the Formula cell becomes enabled and another, empty row is added to the Formula Editor table.
- Drag&Drop some Analyzing Function or User Function from the “Main -> Servers -> *YourServer* -> Libraries” branch of the **MTS Explorer** into the Formula cell.
  - Configure the input arguments of the dropped Analyzing Function/User Function so that all of them are valid. Example:
    - Div(Add(~'Some online Data', 50), 100)

Try it out

➔ Create a few formulas which do not use Analyzing Functions or User Functions.

➔ Create a few formulas which use Analyzing Functions.

➔ Create a few formulas which use User Functions.

➔ Save the Monitoring View.

Ready

At this point you know how to add formulas to the Formula Editor table of the **MVE Standard T001**.

6.30.3 Running of Formulas

Open the saved Monitoring View

- The Formula Editor table of the Monitoring View contains several formulas.

Functionality

- Press the icon within the Run cell of the first formula in order to start the calculation.
  - The status within the Status cell switches to “Started”. In case the status is not “Started” but “Stopped with Error”, the entered formula contains an error and detailed information about the occurred error is found within the system log of the ***X-Tools Server***.
  - Within the **MDS Explorer**, an online data with the name from the Name cell appears. In addition, the calculation result is displayed automatically within the Monitoring View.
- Press the icon within the Run cell of the first formula again in order to stop the calculation.
  - The online data disappears automatically from the **MDS Explorer**.
- All calculations are stopped automatically in case the Monitoring View is closed.
- In case a Monitoring View contains both online and offline data, the ~’’ and $’’ syntax must be used within the Name cell in order to define whether the output data shall be online data or offline data. Examples:
  - ~’ My first Formula’
  - $’ My first Formula’

Try it out

➔ Start all of the formulas which have been created during the previous chapter and visualize their results.

➔ Create a new Monitoring View which contains offline data.

➔ Create formulas which base on offline data.

➔ Run the formulas which base on offline data and visualize their results.

Ready

At this point you know how to run formulas from the Formula Editor table of the **MVE Standard T001**. The formulas can use online or offline data as input and are able to output online or offline data.

The syntax of formulas from the Formula Editor table is identical to the syntax of Analyzing Scripts, but each formula from the Formula Editor table can consist of only one row.

6.31 Passing of Command Line Parameters via a Web Browser

6.31.1 General Description

In this chapter you will learn how to pass command line parameters to ***X-Tools*** via a web browser.

This chapter contains the following topics:

- **Important Facts**
- **Example Web Page**

6.31.2 Important Facts

In order to be able to pass command line parameters to ***X-Tools*** via a web browser, ***X-Tools*** must be registered for the URI scheme “xtools:”. The ***X-Tools Client*** creates this registration automatically during its startup in case the global option “URI Scheme” is configured to “Register the URI Scheme “xtools:” for the X-Tools Client”. On some computers, it may be necessary to start the ***X-Tools Client*** with Windows administrator rights once in order to allow it to create this registration properly.

Basically, all commands which can be passed to the ***X-Tools Client*** via its command line also can be passed via a web browser. The following examples show the HTML code which can be used in order to prepare a link which calls ***X-Tools***:

```
<a href="xtools:/openMVF &quot;YourServer - Example Location://Example for an IFN ANALOGINPUT&quot;">
    Open a Monitoring View
</a>

<a href="xtools:/openMVF &quot;YourServer - Example Location://Example for an IFN ANALOGINPUT&quot;
/fullscreen">
    Open a Monitoring View in full screen mode
</a>

<a href="xtools:/printMVF &quot;YourServer - Example Location://Tutorial\Clock\Clock&quot;">
    Print a Monitoring View
</a>
```

" must be used instead of the “ character around the path to the above Monitoring Views because “ itself is part of the syntax of href already and can not be used inside the href target.

6.31.3 Example Web Page

Prepare the Example Web Page

- Open the file “..\CMS X-Tools Pro\Client\User\Example Files\Examples for X-Tools URI Calls.htm” within an editor like Notepad.
- Replace the three occurrencies of the text “YourServer” through the computer name of your PC.
- Save the .htm file.

Open a Web Browser

- Open the edited .htm file within a web browser.

Functionality

- Start ***X-Tools***.
  - Ensure that the global option “URI Scheme” of the ***X-Tools Client*** is configured to “Register the URI Scheme “xtools:” for the X-Tools Client”.
  - In case the value of “URI Scheme” has been changed, a restart of the ***X-Tools Client*** is necessary in order to take over the new setting.
  - The ***X-Tools Client*** is registered successfully in case its system log contains either no log entry about the URI scheme or it contains a log entry which points out that the registration was successful.

- Prepare the used data:
  - Start the Interface Profile “Simulation T001” from the example location which is delivered together with ***X-Tools***.
  - Start the Device Profile “IFN ANALOGINPUT” from the example location which is delivered together with ***X-Tools***.
  - Start all three Analyzing Models from the “Tutorial\Clock” directory of the example location which is delivered together with ***X-Tools***.
- Click onto the “Open a Monitoring View” link at the opened web page.
  - There may be dialogs popping up which ask whether it shall be allowed to start the ***X-Tools Client***. Allow this access in order to be able to pass the command line parameters to the ***X-Tools Client***.
- The ***X-Tools Client*** comes into the foreground and displays the Monitoring View “Example for an IFN ANALOGINPUT”.
- Click onto the “Open a Monitoring View in full screen mode” link at the opened web page.
  - There may be dialogs popping up which ask whether it shall be allowed to start the ***X-Tools Client***. Allow this access in order to be able to pass the command line parameters to the ***X-Tools Client***.
- The ***X-Tools Client*** comes into the foreground and displays the Monitoring View “Example for an IFN ANALOGINPUT” in full screen mode.
- Click onto the “Print a Monitoring View” link at the opened web page.
  - There may be dialogs popping up which ask whether it shall be allowed to start the ***X-Tools Client***. Allow this access in order to be able to pass the command line parameters to the ***X-Tools Client***.
- The ***X-Tools Server*** prints the Monitoring View “Clock” via the default printer of Windows.

Try it out

➔ Create a web page which contains the links to open a few different Monitoring Views.

➔ Add also links to print a few different Monitoring Views.

➔ Use the links from the created web page in order to open and print the linked Monitoring Views.

Ready

At this point you know how to pass command line parameters to ***X-Tools*** via a web browser.

6.32 Parameters for Configuration Files

6.32.1 General Description

In this chapter you will learn how to configure and to use Configuration Files which provide parameters.

This chapter contains the following topics:

- **Important Facts**
- **Provided Parameters Table**
- **Device Profiles**
- **Monitoring Views**
- **Analyzing Models**
- **Analyzing Scripts**
- **Storage Profiles**

6.32.2 Important Facts

Varios Configuration Files of ***X-Tools*** allow to provide some of their settings as parameter. The value of such a provided parameter can be passed at the moment when the Configuration File is started via the Analyzing Function StartEx() and/or when a new Configuration File shall be created automatically via the Analyzing Function CreateCFEx().

The following types of Configuration Files support provided parameters:

- Device Profiles
- Monitoring Views
- Analyzing Models
- Analyzing Scripts
- Storage Profiles

Some Configuration Files support to pass multiple data names for one provided parameter. Examples of such places are:

- Monitoring Views allow to pass multiple data names for one single legend entry
- Storage Profiles allow to pass multiple data nemas for one single entry of the Data table

In order to pass multiple data names for a single provided parameter, the | separator must be used. Examples:

- Data1|Data2
- Data1|Data2|Data3

Refer to the user manual of each editor for detailed information about which settings support this syntax.

6.32.3 Provided Parameters Table

The Provided Parameters table is available within the editor of each Configuration File which supports provided parameters. This table can be used in order change the (full and short) name, unit, description and default value of each provided parameter.

The following screenshot shows an example of a Provided Parameters table:

Provided Parameters

No.	Full Name	Short Name	Unit	Description	Default Value	Parent Function	Parent Argument
1	Shape	Shape	n/a	form of the wave which shall be generated	Sine	GenWF	Shape
2	Amplitude	Amp	1	amplitude of the waveform which shall be generated	10	GenWF	Amp
3	Frequency	Frq	Hz	frequency of the waveform which shall be generated	0.5	GenWF	Frq
4	Offset	Off	1	steady component of the waveform which shall be generated	2.5	GenWF	Off
5	Phase	Pha	rad	phase-shift of the waveform which shall be generated	0	GenWF	Pha
6	Name	Name	n/a	name of the data or values which shall be written to the system	Default Name	Out	Name

Figure 61: Example of a Provided Parameters Table

Adding of parameters to this table can be dependent to the editor, but typically the Add Parameter context menu item can be used above all settings that can be added as parameter of the Configuration File.

6.32.4 Device Profiles

Open the DMS Explorer

- Open the **Device Management System** and ensure that the **DMS Explorer** is displayed at the left side of the **Device Management System** workspace.

Create a Device Profile which provides Parameters

- Open an existing Device Profile from the example location, e.g. the Device Profile “IFN ANALOGINPUT”.
  - The Device Profile opens within its **Device Profile Editor**. The Provided Parameters table is displayed directly below the Device Profile Settings table.
- Call the context menu above the value of the Data Name Prefix cell of the Device Profile Settings table and choose the Add Parameter context menu item.
  - The Data Name Prefix is added as new entry to the Provided Parameters table.
- Call the context menu above the value of a Name cell of the Device Profile Data table and choose the Add Parameter context menu item.
  - The data name is added as new entry to the Provided Parameters table.
- Save the Device Profile under a new name like “DPF Template”.
  - At this moment, you have created a Device Profile which provides parameters. The Provided Parameters table may look like in the following example:

Provided Parameters

No.	Full Name	Short Name	Unit	Description	Default Value	Reference
1	DataNamePrefix	DtNmPrfx		Data Name Prefix	IFN ANALOGINPUT	Data Name Prefix
2	Data0001	Dt0001		Data 0001	CH2	Data Name

*Figure 62: Example of the Provided Parameters Table of the **DPE IFN ANALOGINPUT***

  - In case the Device Profile is started directly (= via the **DMS Explorer** or via the Analyzing Function Start()), the default values from the Provided Parameters table are being used for all settings that are configured as provided parameter.

Start the Device Profile and pass Parameter Values

- Create a new Analyzing Script within the same location to which also the Device Profile has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function StartEx() in order to start your Device Profile and to pass parameters to it like it is seen in the following example:

```
var main()
{
    // the following command starts the Device Profile and passes parameters to it
    StartEx( "://DPF Template.dpf.xts", "DoNotWait", 0,
             /*Parameter1=*/ "Prefix ",
             /*Parameter2=*/ "Name"; );
}
```

Figure 63: Example of an Analyzing Script which starts a Device Profile with Parameters

Note

In case the Analyzing Script shall start the Device Profile without changing of the default values of the provided parameters, the Analyzing Function Start() must be used instead of StartEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it starts the Device Profile. The values which were specified by the Analyzing Script within its call to StartEx() are taken over by the Device Profile for its current run, but the values within the Device Profile File are not changed.

Use the Device Profile as Template and create another Device Profile automatically

- Create a new Analyzing Script within the same location to which also the Device Profile has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function CreateCFEx() in order to create a new Device Profile with new default values for all provided parameters like it is seen in the following example:

```
var main()
{
    // the following command creates a new Device Profile with new default values
    CreateCFEx( "://DPF Template.dpf.xts",
                "://New Device Profile.dpf.xts",
                /*Parameter1=*/ "Prefix ",
                /*Parameter2=*/ "Name"; );
}
```

Figure 64: Example of an Analyzing Script which creates a new Device Profile out of a Template

Note

In case the Analyzing Script shall copy the Device Profile without changing of the default values of the provided parameters, the Analyzing Function CreateCF() must be used instead of CreateCFEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it creates a new Device Profile. The values which were specified by the Analyzing Script within its call to CreateCFEx() are taken as default values of the provided parameters by the new Device Profile.
- Use the **DMS Explorer** in order to start the Device Profile which has been created automatically.
  - The new Device Profile starts with the default values of its provided parameters which were passed by CreateCFEx().

Try it out

➔ Create a new Device Profile which provides several parameters.

➔ Change the Full Name, Short Name, Unit, Description and Default Value within the Provided Parameters table.

➔ Start the Device Profile via the **DMS Explorer** in order to see which values it uses. Stop the Device Profile afterwards.

➔ Create and start an Analyzing Script which creates a copy of the Device Profile with different default values inside the Provided Parameters table.

➔ Create and start an Analyzing Script which starts the copied Device Profile with again different values for each provided parameter.

Ready

At this point you know how to prepare a Device Profile so that it provides parameters, how to start such a Device Profile with new values and how to create a new Device Profile with new default values automatically. Check the user manual of each **Device Profile Editor** for detailed information about which settings can be configured as provided parameters.

6.32.5 Monitoring Views

Open the MTS Explorer

- Open the **Monitoring System** and ensure that the **MTS Explorer** is displayed at the left side of the **Monitoring System** workspace.

Create a Monitoring View which provides Parameters

- Open an existing Monitoring View from the example location, e.g. the Monitoring View “Example for an IFN ANALOGINPUT”.
  - The Monitoring View opens within its **Monitoring View Editor**. The Provided Parameters table is displayed directly below the Monitoring View Settings table.
- Define a Trigger Data Name within the Automatic Reports Trigger table so that this Monitoring View can be started later.
- Call the context menu above the value of the Data Name Prefix cell of the Monitoring View Settings table and choose the Add Parameter context menu item.
  - The Data Name Prefix is added as new entry to the Provided Parameters table.
- Call the context menu above the value of the Width cell of the Automatic Reports Parameters table and choose the Add Parameter context menu item.
  - The Width is added as new entry to the Provided Parameters table.
- Call the context menu above any legend entry and choose the Add Parameter context menu item.
  - The data name is added as new entry to the Provided Parameters table.
- Save the Monitoring View under a new name like “MVF Template”.
  - At this moment, you have created a Monitoring View which provides parameters. The Provided Parameters table may look like in the following example:

Provided Parameters

No.	Full Name	Short Name	Unit	Description	Default Value	Reference
1	DataNamePrefix	DtNmPrfx		Data Name Prefix		Data Name Prefix
2	Width	Wdth	pixels	Width	1024	Width
3	Data0001	Dt0001		Data 0001	IFN ANALOGINPUT CH1	Monitoring Chart 03\Data0001

*Figure 65: Example of the Provided Parameters Table of the **MVE Standard T001***

  - In case the Monitoring View is started directly (= via the **MTS Explorer** or via the Analyzing Function Start()), the default values from the Provided Parameters table are being used for all settings that are configured as provided parameter.

Start the Monitoring View and pass Parameter Values

- Create a new Analyzing Script within the same location to which also the Monitoring View has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function StartEx() in order to start your Monitoring View and to pass parameters to it like it is seen in the following example:

```
var main()
{
    // the following command starts the Monitoring View and passes parameters to it
    StartEx( "://MVF Template.mvf.xts", "DoNotWait", 0,
             /*Parameter1=*/ "Prefix ",
             /*Parameter2=*/ 1600,
             /*Parameter3=*/ "Name"; );
}
```

Figure 66: Example of an Analyzing Script which starts a Monitoring View with Parameters

Note

In case the Analyzing Script shall start the Monitoring View without changing of the default values of the provided parameters, the Analyzing Function Start() must be used instead of StartEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it starts the Monitoring View. The values which were specified by the Analyzing Script within its call to StartEx() are taken over by the Monitoring View for its current run, but the values within the Monitoring View File are not changed.

Use the Monitoring View as Template and create another Monitoring View automatically

- Create a new Analyzing Script within the same location to which also the Monitoring View has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function CreateCFEx() in order to create a new Monitoring View with new default values for all provided parameters like it is seen in the following example:

```
var main()
{
    // the following command creates a new Monitoring View with new default values
    CreateCFEx( "://MVF Template.mvf.xts",
                "://New Monitoring View.mvf.xts",
                /*Parameter1=*/ "Prefix ",
                /*Parameter2=*/ "Name"; );
}
```

Figure 67: Example of an Analyzing Script which creates a new Monitoring View out of a Template

Note

In case the Analyzing Script shall copy the Monitoring View without changing of the default values of the provided parameters, the Analyzing Function CreateCF() must be used instead of CreateCFEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it creates a new Monitoring View. The values which were specified by the Analyzing Script within its call to CreateCFEx() are taken as default values of the provided parameters by the new Monitoring View.
- Use the **MTS Explorer** in order to start the Monitoring View which has been created automatically.
  - The new Monitoring View starts with the default values of its provided parameters which were passed by CreateCFEx().

Try it out

➔ Create a new Monitoring View which provides several parameters.

➔ Change the Full Name, Short Name, Unit, Description and Default Value within the Provided Parameters table.

➔ Start the Monitoring View via the **MTS Explorer** in order to see which values it uses. Stop the Monitoring View afterwards.

➔ Create and start an Analyzing Script which creates a copy of the Monitoring View with different default values inside the Provided Parameters table.

➔ Create and start an Analyzing Script which starts the copied Monitoring View with again different values for each provided parameter.

Ready

At this point you know how to prepare a Monitoring View so that it provides parameters, how to start such a Monitoring View with new values and how to create a new Monitoring View with new default values automatically. Check the user manual of the **Monitoring View Editor** and of each **Monitoring Chart** for detailed information about which settings can be configured as provided parameters.

6.32.6 Analyzing Models

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.

Create an Analyzing Model which provides Parameters

- Open an existing Analyzing Model from the example location, e.g. the Analyzing Model “Arithmetic\Abs 01”.
  - The Analyzing Model opens within its **Analyzing Model Editor**. The Provided Parameters table is displayed directly below the Analyzing Model Settings table.
- Call the context menu above the value of the Data Name Prefix of Output Data cell of the Analyzing Model Settings table and choose the Add Parameter context menu item.
  - The Data Name Prefix of Output Data is added as new entry to the Provided Parameters table.
- Call the context menu above the input argument of any Analyzing Function and choose the Add Parameter context menu item.
  - The input argument is added as new entry to the Provided Parameters table.
- Save the Analyzing Model under a new name like “AMF Template”.
  - At this moment, you have created an Analyzing Model which provides parameters. The Provided Parameters table may look like in the following example:

Provided Parameters

No.	Full Name	Short Name	Unit	Description	Default Value	Parent Function	Parent Argument	Sequence ID
1	PrefixOutputData	PrfxOutDt			Abs 01	N/A	N/A	N/A
2	Amplitude	Amp	1	amplitude of the waveform which shall be generated	1	GenWF	Amp	0

*Figure 68: Example of the Provided Parameters Table of the **AME Standard T001***

  - In case the Analyzing Model is started directly (= via the **ANS Explorer** or via the Analyzing Function Start()), the default values from the Provided Parameters table are being used for all settings that are configured as provided parameter.

Start the Analyzing Model and pass Parameter Values

- Create a new Analyzing Script within the same location to which also the Analyzing Model has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function StartEx() in order to start your Analyzing Model and to pass parameters to it like it is seen in the following example:

```
var main()
{
    // the following command starts the Analyzing Model and passes parameters to it
    StartEx( "://AMF Template.amf.xts", "DoNotWait", 0,
             /*Parameter1=*/ "Prefix ",
             /*Parameter2=*/ 10; );
}
```

Figure 69: Example of an Analyzing Script which starts an Analyzing Model with Parameters

Note

In case the Analyzing Script shall start the Analyzing Model without changing of the default values of the provided parameters, the Analyzing Function Start() must be used instead of StartEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it starts the Analyzing Model. The values which were specified by the Analyzing Script within its call to StartEx() are taken over by the Analyzing Model for its current run, but the values within the Analyzing Model File are not changed.

Use the Analyzing Model as Template and create another Analyzing Model automatically

- Create a new Analyzing Script within the same location to which also the Analyzing Model has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function CreateCFEx() in order to create a new Analyzing Model with new default values for all provided parameters like it is seen in the following example:

```
var main()
{
    // the following command creates a new Analyzing Model with new default values
    CreateCFEx( "://AMF Template.amf.xts",
                "://New Analyzing Model.amf.xts",
                /*Parameter1=*/ "Prefix ",
                /*Parameter2=*/ 10; );
}
```

Figure 70: Example of an Analyzing Script which creates a new Analyzing Model out of a Template

Note

In case the Analyzing Script shall copy the Analyzing Model without changing of the default values of the provided parameters, the Analyzing Function CreateCF() must be used instead of CreateCFEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it creates a new Analyzing Model. The values which were specified by the Analyzing Script within its call to CreateCFEx() are taken as default values of the provided parameters by the new Analyzing Model.
- Use the **ANS Explorer** in order to start the Analyzing Model which has been created automatically.
  - The new Analyzing Model starts with the default values of its provided parameters which were passed by CreateCFEx().

Try it out

➔ Create a new Analyzing Model which provides several parameters.

➔ Change the Full Name, Short Name, Unit, Description and Default Value within the Provided Parameters table.

➔ Start the Analyzing Model via the **ANS Explorer** in order to see which values it uses. Stop the Analyzing Model afterwards.

➔ Create and start an Analyzing Script which creates a copy of the Analyzing Model with different default values inside the Provided Parameters table.

➔ Create and start an Analyzing Script which starts the copied Analyzing Script with again different values for each provided parameter.

Ready

At this point you know how to prepare an Analyzing Model so that it provides parameters, how to start such an Analyzing Model with new values and how to create a new Analyzing Model with new default values automatically. Check the user manual of the **Analyzing Model Editor** for detailed information about which settings can be configured as provided parameters.

6.32.7 Analyzing Scripts

Open the ANS Explorer

- Open the **Analyzing System** and ensure that the **ANS Explorer** is displayed at the left side of the **Analyzing System** workspace.

Create an Analyzing Script which provides Parameters

- Open an existing Analyzing Script from the example location, e.g. the Analyzing Script “Arithmetic\Abs 01”.
  - The Analyzing Script opens within its **Analyzing Script Editor**. The Provided Parameters table is displayed directly below the Analyzing Script Settings table.

- Call the context menu above the value of the Data Name Prefix of Output Data cell of the Analyzing Script Settings table and choose the Add Parameter context menu item.
  - The Data Name Prefix of Output Data is added as new entry to the Provided Parameters table.
- Double-click into the Name cell of the last, empty row of the Provided Parameters table in order to add another parameter. Set the following values:
  - Full Name = Amplitude
  - Short Name = Ampl
  - Unit = “”
  - Description = “”
  - Default Value = 20
- Use the newly defined parameter (short name = Ampl) within the actual Analyzing Script like it is shown in the following example:

```
var main()
{
    var dSignal = GenWF("Sine",%'Ampl',1,2);
    ~'Signal' = dSignal;
    ~'AbsForSignal' = Abs(dSignal);
}
```

Figure 71: Example of an Analyzing Script which uses a provided Parameter

- Save the Analyzing Script under a new name like “ASF Template”.
  - At this moment, you have created an Analyzing Script which provides parameters. The Provided Parameters table may look like in the following example:

Provided Parameters

No.	Full Name	Short Name	Unit	Description	Default Value
1	PrefixOutputData	PrfxOutDt			Abs 01
2	Amplitude	Ampl			20
3	Enter a new full name in order to define another parameter				

*Figure 72: Example of the Provided Parameters Table of the **ASE Standard T001***

  - In case the Analyzing Script is started directly (= via the **ANS Explorer** or via the Analyzing Function Start()), the default values from the Provided Parameters table are being used for all settings that are configured as provided parameter.

Start the Analyzing Script and pass Parameter Values

- Create a new Analyzing Script within the same location to which also the template Analyzing Script has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function StartEx() in order to start your template Analyzing Script and to pass parameters to it like it is seen in the following example:

```
var main()
{
    // the following command starts the Analyzing Script and passes parameters to it
    StartEx( "://AsF Template.asf.xts", "DoNotWait", 0,
             /*Parameter1=*/ "Prefix ",
             /*Parameter2=*/ 30; );
}
```

Figure 73: Example of an Analyzing Script which starts an Analyzing Script with Parameters

Note

In case the Analyzing Script shall start the template Analyzing Script without changing of the default values of the provided parameters, the Analyzing Function Start() must be used instead of StartEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it starts the template Analyzing Script. The values which were specified by the Analyzing Script within its call to StartEx() are taken over by the template Analyzing Script for its current run, but the values within the template Analyzing Script File are not changed.

Use the Analyzing Script as Template and create another Analyzing Script automatically

- Create a new Analyzing Script within the same location to which also the template Analyzing Script has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function CreateCFEx() in order to create a new Analyzing Script with new default values for all provided parameters like it is seen in the following example:

```
var main()
{
    // the following command creates a new Analyzing Script with new default values
    CreateCFEx( "://ASF Template.aSf.xts",
                "://New Analyzing Script.asf.xts",
                /*Parameter1=*/ "Prefix ",
                /*Parameter2=*/ 40; );
}
```

Figure 74: Example of an Analyzing Script which creates a new Analyzing Script out of a Template

Note

In case the Analyzing Script shall copy the template Analyzing Script without changing of the default values of the provided parameters, the Analyzing Function CreateCF() must be used instead of CreateCFEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it creates a new Analyzing Script. The values which were specified by the Analyzing Script within its call to CreateCFEx() are taken as default values of the provided parameters by the new Analyzing Script.
- Use the **ANS Explorer** in order to start the Analyzing Script which has been created automatically.
  - The new Analyzing Script starts with the default values of its provided parameters which were passed by CreateCFEx().

Try it out

➔ Create a new Analyzing Script which provides several parameters.

➔ Change the Full Name, Short Name, Unit, Description and Default Value within the Provided Parameters table.

➔ Start the Analyzing Script via the **ANS Explorer** in order to see which values it uses. Stop the Analyzing Script afterwards.

➔ Create and start an Analyzing Script which creates a copy of the template Analyzing Script with different default values inside the Provided Parameters table.

➔ Create and start an Analyzing Script which starts the copied Analyzing Script with again different values for each provided parameter.

Ready

At this point you know how to prepare an Analyzing Script so that it provides parameters, how to start such an Analyzing Script with new values and how to create a new Analyzing Scroüt with new default values automatically. Check the user manual of the **Analyzing Script Editor** for detailed information about which settings can be configured as provided parameters.

6.32.8 Storage Profiles

Open the STS Explorer

- Open the **Storage System** and ensure that the **STS Explorer** is displayed at the left side of the **Storage System** workspace.

Create a Storage Profile which provides Parameters

- Open an existing Storage Profile from the example location, e.g. the Storage Profile “Example for an ION AnalogInput T001”.
  - The Storage Profile opens within its **Storage Profile Editor**. The Provided Parameters table is displayed directly below the Storage Profile Settings table.
- Call the context menu above the value of the Data Name Prefix cell of the Storage Profile Settings table and choose the Add Parameter context menu item.
  - The Data Name Prefix is added as new entry to the Provided Parameters table.
- Call the context menu above the value of a Name cell of the Data table and choose the Add Parameter context menu item.
  - The data name is added as new entry to the Provided Parameters table.
- Save the Storage Profile under a new name like “SPF Template”.
  - At this moment, you have created a Storage Profile which provides parameters. The Provided Parameters table may look like in the following example:

Provided Parameters

No.	Full Name	Short Name	Unit	Description	Default Value	Reference
1	DataNamePrefix	DtNmPrfx		DataNamePrefix		Data Name Prefix
2	Data0001	Dt0001		Data 0001	ION AnalogInput T001 CH06	Data Name

*Figure 75: Example of the Provided Parameters Table of the **SPE Binary T001***

  - In case the Stroage Profile is started directly (= via the **STS Explorer** or via the Analyzing Function Start()), the default values from the Provided Parameters table are being used for all settings that are configured as provided parameter.

Start the Storage Profile and pass Parameter Values

- Create a new Analyzing Script within the same location to which also the Storage Profile has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function StartEx() in order to start your Storage Profile and to pass parameters to it like it is seen in the following example:

```
var main()
{
    // the following command starts the Storage Profile and passes parameters to it
    StartEx( "://SPF Template.spf.xts", "DoNotWait", 0,
             /*Parameter1=*/ "ION AnalogInput T001 ",
             /*Parameter2=*/ "CH01"; );
}
```

Figure 76: Example of an Analyzing Script which starts a Storage Profile with Parameters

Note

In case the Analyzing Script shall start the Storage Profile without changing of the default values of the provided parameters, the Analyzing Function Start() must be used instead of StartEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it starts the Storage Profile. The values which were specified by the Analyzing Script within its call to StartEx() are taken over by the Storage Profile for its current run, but the values within the Storage Profile File are not changed.

Use the Storage Profile as Template and create another Storage Profile automatically

- Create a new Analyzing Script within the same location to which also the Storage Profile has been stored.
- Change the Operating Mode to “Sequential”.
- Use the Analyzing Function CreateCFEx() in order to create a new Storage Profile with new default values for all provided parameters like it is seen in the following example:

```
var main()
{
    // the following command creates a new Storage Profile with new default values
    CreateCFEx( "://SPF Template.spf.xts",
                "://New Storage Profile.spf.xts",
                /*Parameter1=*/ "ION AnalogInput T001 ",
                /*Parameter2=*/ "CH02"; );
}
```

Figure 77: Example of an Analyzing Script which creates a new Storage Profile out of a Template

Note

In case the Analyzing Script shall copy the Storage Profile without changing of the default values of the provided parameters, the Analyzing Function CreateCF() must be used instead of CreateCFEx().

- Save and start the Analyzing Script.
  - The Analyzing Script starts and it creates a new Storage Profile. The values which were specified by the Analyzing Script within its call to CreateCFEx() are taken as default values of the provided parameters by the new Storage Profile.
- Use the **STS Explorer** in order to start the Storage Profile which has been created automatically.
  - The new Storage Profile starts with the default values of its provided parameters which were passed by CreateCFEx().

Try it out

➔ Create a new Storage Profile which provides several parameters.

➔ Change the Full Name, Short Name, Unit, Description and Default Value within the Provided Parameters table.

➔ Start the Storage Profile via the **STS Explorer** in order to see which values it uses. Stop the Storage Profile afterwards.

➔ Create and start an Analyzing Script which creates a copy of the Storage Profile with different default values inside the Provided Parameters table.

➔ Create and start an Analyzing Script which starts the copied Storage Profile with again different values for each provided parameter.

Ready

At this point you know how to prepare a Storage Profile so that it provides parameters, how to start such a Storage Profile with new values and how to create a new Storage Profile with new default values automatically. Check the user manual of the **Storage Profile Editor** for detailed information about which settings can be configured as provided parameters.

6.33 Startup Analyzing Scripts

6.33.1 General Description

In this chapter you will learn how to use startup Analyzing Scripts in order to start all Configuration Files in a defined order during each startup of the ***X-Tools Server***.

This chapter contains the following topics:

- **Important Facts**
- **Startup Analyzing Script**

6.33.2 Important Facts

Per default, the ***X-Tools Server*** remembers which Configuration Files were started at the moment of its termination and it automatically restarts these Configuration Files during its next startup.

Although this is a very convenient functionality without any additional maintenance needs for a lot of applications, there also may be applications for which a defined order for the startup of Configuration Files is significantly better. An example for such an application are configurations which base on sequence control Analyzing Scripts – such configurations typically start and stop a lot of Configuration Files by themselves and automatically, thus it would not make sense to restart the Configuration Files which were running at a random moment (because the ***X-Tools Server*** could be terminated at any time).

In order to use startup Analyzing Scripts instead of the default startup files, the global option “Startup Type” must be configured to “Use Startup Analyzing Scripts for the Startup of Configuration Files during the Startup of the X-Tools Server”. The global option “Startup Analyzing Script” can be used in order to specify the name of the Analyzing Script which shall be started as startup Analyzing Script by the ***X-Tools Server*** during its own startup.

The ***X-Tools Server*** tries to start the specified Analyzing Scripts from the root directory of each present location, thus it is possible (but not necessary) to have more than one startup Analyzing Scripts (= one within each present location).

6.33.3 Startup Analyzing Script

The startup Analyzing Script must be running in the operating mode “Sequential” in order to be able to startup other Configuration Files. The following example shows the startup Analyzing Script which is delivered as part of the example location together with ***X-Tools***:

```
var main()
{
    // turn on the simulated devices
    Start( "://Simulation T001.ipf.xts", "DoNotWait" );

    // turn on some Device Profiles
    Start( "://IFN ANALOGINPUT.dpf.xts", "DoNotWait" );
    Start( "://IFN VIB-ACC.dpf.xts", "DoNotWait" );
    Start( "://ION AnalogInput T001.dpf.xts", "DoNotWait" );
    Start( "://ION SIMATIC S7-1200.dpf.xts", "DoNotWait" );
    Start( "://ION SIMATIC S7 PN.dpf.xts", "DoNotWait" );

    // turn on some online analysis
    Start( "://Spectral/FFT 01.amf.xts", "DoNotWait" );
    Start( "://Statistic/H1D 01.asf.xts", "DoNotWait" );

    // turn on some Storage Profiles
    Start( "://Example for an ION AnalogInput T001.spf.xts", "DoNotWait" );
    Start( "://Example for an ION SIMATIC S7 PN.spf.xts", "DoNotWait" );

    // load some offline data
    LoadData( "X-Tools Offline Data Files://ION AnalogInput T001" );

    // turn on some offline analysis and wait until it has finished
    Start( "://Arithmetic/Abs 02.amf.xts", "Wait" );
}
```

Figure 78: Example of a Startup Analyzing Script

Try it out

- ➔ Ensure that the directory "..\Server\User\Example Location" which is delivered together with ***X-Tools*** is present as Configuration File location at the ***X-Tools Server***.
- ➔ Start multiple Configuration Files from this location.
- ➔ Ensure that the directory "..\Server\User\Example Offline Files\X-Tools Offline Data Files" which is delivered together with ***X-Tools*** is present as offline data location at the ***X-Tools Server***. The symbolic name of this offline data location must be "X-Tools Offline Data Files".
- ➔ Configure the global option "Startup Type" to "Use Startup Analyzing Scripts for the Startup of Configuration Files during the Startup of the X-Tools Server".
- ➔ Leave the default value of the global option "Startup Analyzing Script" (= "Startup").
- ➔ Terminate the ***X-Tools Server***.
- ➔ Start the ***X-Tools Server***.

  The ***X-Tools Server*** does not start the Configuration Files which were running at the moment of its termination. Instead, it starts only the Analyzing Script "Startup" from the example location and this Analyzing Script starts several other Configuration Files.

Ready

At this point you know how to use startup Analyzing Scripts in order to configure which Configuration Files shall be started in which order during the startup of the ***X-Tools Server***.

6.34 Supported CSV (*.csv) Format for Data

6.34.1 General Description

In this chapter you will learn about the details of the file format which is being used for exporting and importing of CSV (*.csv) data files.

This chapter contains the following topics:

- **Exporting of Data to CSV Files**
- **Importing of Data from CSV Files**
- **CSV File Format for Data**

6.34.2 Exporting of Data to CSV Files

Open the ODL Standard T001

- Open the **Storage System** and ensure that the **ODL Standard T001** is displayed within the **Storage System** workspace. The **ODL Standard T001** must contain at least one offline file location of type “X-Tools Offline Data Files (*.sdf.xts)” and at least one offline file location of type “CSV Files (*.csv)”.
- Ensure that you have several data available below the “Main -> Servers -> *YourServer* -> Offline Data -> All Offline Data” branch of the **MDS Explorer**.

Functionality

- Method 1 for exporting of data to CSV files:
  - Within the **ODL Standard T001**, call the context menu from a directory or file below an offline file location of type “X-Tools Offline Data Files (*.sdf.xts)”.
  - Choose Export or Advanced Export… from the context menu in order to start the export operation.
- Method 2 for exporting of data to CSV files:
  - Within the **MDS Explorer**, call the context menu from an offline data category or from an offlne data.
  - Choose Export or Advanced Export… from the context menu in order to start the export operation.
- The exported CSV file is put into the chosen/last used offline file location of type “CSV Files (*.csv)”.
- All types of offline data (simple data types, 1-dimensional buffers, 2-dimensional buffers, spectrums, …) can be exported to CSV files.

Try it out

➔ Export some offline data files (simple data types and extended data types) via the **ODL Standard T001**.

➔ Export some offline data (simple data types and extended data types) via the **MDS Explorer**.

Ready

At this point you know how to export CSV data.

6.34.3 Importing of Data from CSV Files

Open the ODL Standard T001

- Open the **Storage System** and ensure that the **ODL Standard T001** is displayed within the **Storage System** workspace. The **ODL Standard T001** must contain at least one offline file location of type “CSV Files (*.csv)”.

Functionality

- Within the **ODL Standard T001**, call the context menu from a directory or file below an offline file location of type “CSV Files (*.csv)”.
- Choose Append, Overwrite, Advanced Append.. or Advanced Overwrite… from the context menu in order to start the import operation.

- Currently, only simple data types (Booleans, Integers, decimals and strings) can be imported from CSV files.

Try it out

➔ Export some offline data (simple data types) via the **MDS Explorer**.

➔ Use a text editor in order to change an exported CSV file.

➔ Load the modified CSV file via the **ODL Standard T001**.

➔ Compare the original offline data and the offline data from the CSV file within the **Monitoring System**. Both data should be identical with exception of the part which has been modified within the text editor.

Ready

At this point you know how to modify and import CSV data.

6.34.4 CSV File Format for Data

Boolean, Integer and decimal Types

Each CSV file has three header rows:

- The first row contains the name and description of the meta information of the file, like the name of the data, unit of the data, start date, stop date, the number of contained values and the precision of the data.
- The second row contains the actual values of this meta information (= the actual name, unit, start date, stop date, number of values, precision, … of the contained data).
- The third row contains the name and description of the actual values within the file, like the timestamp and the value.

Example of typical header rows:

```
Data Name;Data Unit;Start Date [ns];Stop Date [ns];Number of Values;Precision
My secret Data;GV;1194453433688958640;1194453433688959640;2;Decimal32
Timestamp [ns];Value
```

The format of the timestamp can be one of the following:

- absolute or relative time in ns since 1970-01-01 00:00:00.000.000.000 GMT
  - example:
    - Timestamp [ns];Value
      1194453433688958640;-2.36511230e-003
  - During the import, all data which contain timestamps in ns between -315576000 … +315576000 (~ -10 years … + 10 years) are considered to be data with relative timestamps and all data which contain only timestamps which are smaller or bigger are considered to be data with absolute timestamps.
- absulte timestamp in textual form
  - example:
    - Timestamp [Date];Timestamp [Time];Timestamp [ns];Value
      07.11.2007;17:37:13;688958640;-2.36511230e-003
- relative timestamp in textual form
  - example:
    - Timestamp [Days];Timestamp [Time];Timestamp [ns];Value
      0;00:00:00;123000000;-2.36511230e-003

String Types

Compared to Boolean, Integer and decimal types, strings contain the folloing additional columns within their first two header rows:

- Size
  - size of one value of the string, in bytes
- Type
  - currently, the “Type” of strings is always “string”
- Format
  - currently, the “Format” of strings is always 1
- Code Page
  - contains the code page of the string which it had a the moment when it has been exported

Example of typical header rows:

```
Data Name;Data Unit;Start Date [ns];Stop Date [ns];Number of Values;Precision;Size;Type;Format;Code Page
My secret Data;;1194453433688958640;1194453433688959640;2;String;16;string;1;1252
Timestamp [ns];Value
```

Extended Data Types

Extended data types are formats like 1-dimensional buffers, 2-dimensional buffers or spectrums. Their export format is principially identical to the export format of simple data types, but of course the header rows contain other and/or additional information. In addition, also the data rows provide another format which is more suitable for the representation of each extended data type.

Extended data types can be exported to but can not be imported from CSV files by the current version of ***X-Tools***. In addition, it also is not possible at the moment to export data of type SpectrumEnvelope.

6.35 File Associations

6.35.1 General Description

In this chapter you will learn about the types of files which can be opened in ***X-Tools*** via a double-click onto the file within the Explorer of Windows.

This chapter contains the following topics:

- **Important Facts**
- **Opening of Files via the Explorer of Windows**

6.35.2 Important Facts

The global options of the ***X-Tools Client*** contain the parameter “X-Tools Files” within the “File Associations” group. In case the value of this parameter is “Associate X-Tools Files with the X-Tools Client”, the ***X-Tools Client*** ensures during each startup that the following file types are associated to it through Windows:

- *.xtc
- *.xts
- *.wav

After a double-click onto an associated file within the Explorer of Windows, the ***X-Tools Client*** is started through Windows and starts to open the chosen files. In case the ***X-Tools Client*** was running already before the double-click, the already present ***X-Tools Client*** opens the chosen files and no new ***X-Tools Client*** is started.

6.35.3 Opening of Files via the Explorer of Windows

Open the Explorer of Windows

- Start ***X-Tools*** and terminate afterwards in order to ensure that ***X-Tools*** can configure the file associations.
- Ensure that neither the ***X-Tools Server*** nor the ***X-Tools Client*** are running.
- Open the Explorer of Windows.

Functionality

- The following types of files can be opened by the ***X-Tools Client*** directly (without the ***X-Tools Server***):
  - Main Profiles
  - Interface Profiles
  - Device Profiles
  - Monitoring Views
  - Analyzing Models
  - Analyzing Scripts
  - Storage Profiles
  - Loading Profiles
  - System Log Files
  - Global Options Files
  - User Accounts Files

- The following types of files can be opened by the ***X-Tools Client*** only when the local ***X-Tools Server*** is connected to the ***X-Tools Client***:
  - offline data files (*.sdf.xts)
  - WAVE files (*.wav)
  - User Function Files
  - current log file of the ***X-Tools Server***
- All other types of files can not be opened within a dialog of ***X-Tools***.
- Whenever a file of the ***X-Tools Server*** (*.xts) shall be opened, the ***X-Tools Client*** automatically attempts to connect to the ***X-Tools Server***.
  - In case the connection succeeds and the selected files are available to the ***X-Tools Server*** (= they are present within a known location of the ***X-Tools Server***) they are opened via the ***X-Tools Server***.
  - In case the connection can not be established or the selected files are not available to the ***X-Tools Server***, the opening is either canceled (in case of files which can not be opened without the ***X-Tools Server***) or the user can choose whether the files shall be opened from the local file system of the ***X-Tools Client***.

Try it out

➔ Double-click onto an offline data file (*.sdf.xts) which is present within an offline file location of the ***X-Tools Server***.

The offline data is loaded through ***X-Tools*** and is displayed within the **Monitoring System**.

➔ Double-click onto a WAVE file (*.wav) which is present within an offline file location of the ***X-Tools Server***.

The offline data is loaded through ***X-Tools*** and is displayed within the **Monitoring System**.

➔ Double-click onto an Interface Profile (*.ipf.xts), Analyzing Model (*.amf.xts) or onto another Configuration File which is present within a Configuration File location of the ***X-Tools Server***.

The Configuration File is opened through ***X-Tools*** and is displayed within its according editor.

➔ Double-click onto a system log file (*.slf.xtc or *.slf.xts).

The system log file is opened through ***X-Tools*** and is displayed within the **System Log Viewer**.

Ready

At this point you know how to open offline data files and Configuration Files of ***X-Tools*** via a double-click onto the files within the Explorer of Windows.

7 Attachment A – Best Practices

7.1 Inside Operating Systems

7.1.1 General

Each of the supported operating systems shows slightly different behavior in different situations. At different places, this behavior must be considered when using of ***X-Tools***. The following topics describe the key places of each operating system:

- Microsoft Windows XP
- Microsoft Windows Server 2003
- Microsoft Windows 7

In addition, the following remarks are valid for all operating systems:

- All components of ***X-Tools*** must run with administrator privileges in order to provide all of the required functionality.
  - The behavior of ***X-Tools*** is undefined in case the necessary administrator privileges are not available.
- In case the ***X-Tools Server*** has been started through the ***X-Tools Service*** (either because of the configured automated startup or because of the configured automated restart), the ***X-Tools Server*** runs in session 0.
- Multiple instances of the ***X-Tools Tray*** application can be running under each operating system, but there can be not more than one instance of the ***X-Tools Tray*** per Windows session.
- The current Windows session of each running process can be seen within the “Session ID” column of the “Processes” tab of the Task Manager of Windows.

7.1.2 Microsoft Windows XP

Only one user can be logged in at Microsoft Windows XP at any time. In case the option “Use Fast User Switching” is turned on within the user accounts setting of Windows, multiple Windows sessions can be present simultaneously, but still there can be only one user working with the system at any time.

In case the ***X-Tools Server*** has been started by the ***X-Tools Service*** it is running in session 0. When the ***X-Tools Server*** has been started through the ***X-Tools Service*** it keeps running even in case the user logs off from session 0, because session 0 is never closed.

Session 0 is the local console of Windows XP. It can be reached by logging in to the machine via its local keyboard or by using of the “/admin” or “/console” command line parameter of Remote Desktop.

7.1.3 Microsoft Windows Server 2003

Multiple users can be logged in at and work within one Microsoft Windows Server 2003 machine simultaneously, where each logged in user has his own Windows session.

In case the ***X-Tools Server*** has been started by the ***X-Tools Service*** it is running in session 0. When the ***X-Tools Server*** has been started through the ***X-Tools Service*** it keeps running even in case the user logs off from session 0, because session 0 is never closed.

Session 0 is the local console of Windows Server 2003. It can be reached by logging in to the machine via its local keyboard or by using of the “/admin” or “/console” command line parameter of Remote Desktop.

Note

By default, hardware acceleration for the visualization of graphics is disabled for Windows Server 2003. In order to enable this hardware acceleration, open the “Display Properties” dialog of Windows, switch to the “Settings” tab and click onto the “Advanced” button. Switch to the “Troubleshoot” tab and move the slider to “Full” before you apply your changes via pressing of the “OK” button.

After DirectX has been installed through the setup, “dxdiag.exe” must be started from the DirectX installation directory. Switch to the “Display” tab and configure “DirectDraw Acceleration”, “Direct3D Acceleration” and “AGP Texture Acceleration” all to “Enabled”.

7.1.4 Microsoft Windows 7

Only one user can be logged in at Microsoft Windows 7 at any time. Multiple Windows sessions can be present simultaneously, but there can be only one user working with the system at any time.

In case the ***X-Tools Server*** has been started by the ***X-Tools Service*** it is running in session 0. In case of Windows 7, session 0 is reserved for the system and no user can log in to this session.

In order to allow the system to start the ***X-Tools Server*** automatically, the UAC (User Accounts Control) prompt must be disabled. The following steps can be performed in order to disable the UAC prompt:

- Open “Control Panel” -> “User Accounts”.
- Choose “Change User Account Control settings”.
- Configure the setting to “Never notify” and press “OK”.

In case the UAC prompt is not disabled it will pop up whenever the ***X-Tools Server*** should be started automatically. In this case the ***X-Tools Server*** won’t be started until a user logs in to the computer and acknowledges the UAC prompt manually.

Note

By default, all notification icons of applications are hidden in Windows 7 when they are inactive. This applies also to the task tray icon which is provided through the ***X-Tools Tray*** application.

In case the task tray icon of the ***X-Tools Tray*** application shall be visible permanently, press the “Show hidden Icons” button from the task bar of Windows and press “Customize…”. Within the upcoming dialog, the behavior of the task tray icon of the ***X-Tools Tray*** can be configured to any of the available settings and press “OK” afterwards.

Microsoft Windows 7 is delivered with two versions of the basic IEEE1394 bus driver: The so-called “new” driver with advanced support for IEEE1394b devices and the so-called “legacy” driver. In case the new driver is installed and the IEEE1394 devices are not detected or operating properly, the legacy driver must be installed. The legacy driver also must be installed in case the used IEEE1394 host adapter uses an IEEE1394-1995 PHY. In order to install the legacy driver, the following steps must be performed:

- Open the Device Manager of Windows.
- Expand the “IEEE 1394 Bus host controllers” item.
- Call the context menu of your IEEE1394 host adapter and choose “Update Driver Software…”.
- Select “Browse my computer for driver software”.
- Select “Let me pick from a list of device driver on my computer” and check the box before “Show compatible hardware”.
- Select “1394 OHCI Compliant Host Controller (Legacy)” and click “Next” in order to install the legacy driver.

7.2 Time Synchronization

7.2.1 General

In case ***X-Tools Server*** and ***X-Tools Client*** are running on different computers, it is strongly recommended to use time synchronization for the system times of these computers. Without time synchronization, the system times of the used computers may be different and as a result the visualization within the **Monitoring System** also would be shifted into the past or into the future.

The easiest way to set up time synchronization on a Windows system is to use the built-in w32time service. In the following there is an example, how to set up a Windows time server on a Windows XP machine and how to configure the client computers. Prinzipially the same configurations are valid also for other operating systems like Windows Server 2003 and Windows 7.

7.2.2 Setup a Windows XP Time Server

In order to enable the time server functionality on a Windows XP machine, it is only necessary to change and create some registry entries and to restart the Windows time service afterwards.

The following table contains the registry keys which must be updated in order to setup up the time server. The registry keys are found below “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\...”:

Registry Key	New Value [Dec]
…\Config\AnnounceFlags	**5**
…\TimeProviders\NtpServer\Enabled	**1**
…\TimeProviders\NtpServer\InputProvider	**1**

The above values are examples which may be acceptable for a large range of applications. In case more specific settings are needed, detailed information about the exact meaning of each registry key is found within the documentation about the windows time service which is available directly from Microsoft.

7.2.3 Setup a Windows XP Time Client

In order to enable the time synchronization on a Windows XP machine as time client, it is only necessary to change some registry entries and to restart the Windows time service afterwards.

The following table contains the registry keys which must be updated in order to setup up the time client. The registry keys are found below “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\...”:

Registry Key	New Value [Dec]
…\Config\FrequencyCorrectRate	**4**
…\Config\LargePhaseOffset	**1280000**
…\Config\HoldPeriod	**5**
…\Config\MaxAllowedPhaseOffset	**300**
…\Config\PhaseCorrectRate	**7**
…\Config\SpikeWatchPeriod	**90**
…\Config\UpdateInterval	**30000**
…\Parameters\NtpServer	***YourNTPServerName*****,0x1**
…\Parameters\Type	**NTP**
…\TimeProviders\NtpClient\Enabled	**1**
…\TimeProviders\NtpClient\InputProvider	**1**
…\TimeProviders\NtpClient\SpecialPollInterval	**60**

The above values are examples which may be acceptable for a large range of applications. In case more specific settings are needed, detailed information about the exact meaning of each registry key is found within the documentation about the windows time service which is available directly from Microsoft.

After the values of the Windows time service have been changed within the registry, the Windows time service can be restarted via the following command line:

- net stop w32time && net start w32time

It is suggested to manually synchronize the system time to the time of the time server once after the time service has been configured. The manual synchronization avoids long-lasting synchronization of the system time in case the difference between the system time and the time of the time server is big. The following command line can be executed in order to change the current system time immediately to the time of the time server which is configured for the Windows time service:

- net time /set /yes

The current time difference between the system time and the time of the time server can be visualized and tracked via the following command line:

- w32tm /stripchart /computer:*YourNTPServerName*

7.2.4 Setup of a Computer which is Time Client and Time Server simultaneously

In case the configured time server shall synchronize its time to a global time server (like “time.windows.com”) also, the setup of a time client also must be applied at the computer which runs the time server. In this case, this computer acts as time server for the local time clients and in addition it is a time client which receives its time from e.g. “time.windows.com”.

The following table contains the registry keys which must be updated in order to setup up a computer to act as time server (for other local computers) and as time client (e.g. to “time.microsoft.com”) simultaneously. The registry keys are found below “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W32Time\...”:

Registry Key	New Value [Dec]
…\Config\AnnounceFlags	**5**
…\Config\FrequencyCorrectRate	**4**
…\Config\LargePhaseOffset	**1280000**
…\Config\HoldPeriod	**5**
…\Config\MaxAllowedPhaseOffset	**300**
…\Config\PhaseCorrectRate	**7**
…\Config\SpikeWatchPeriod	**90**
…\Config\UpdateInterval	**30000**
…\Parameters\NtpServer	***YourNTPServerName*,0x1**
…\Parameters\Type	**NTP**
…\TimeProviders\NtpClient\Enabled	**1**
…\TimeProviders\NtpClient\InputProvider	**1**
…\TimeProviders\NtpClient\SpecialPollInterval	**60**
…\TimeProviders\NtpServer\Enabled	**1**
…\TimeProviders\NtpServer\InputProvider	**1**

Depending to the location of the computer, other time servers may be more suitable than “time.windows.com”. A list of available time servers can be found at http://support.ntp.org/bin/view/Servers/StratumTwoTimeServers and at other locations in the Internet.

7.2.5 Microsoft Hotfix 951531

When you start a Windows XP based computer, the Windows time service (W32Time) obtains the current time from the CMOS clock and sets this time to the system time. This system time is synchronized with the time server by the W32Time service. Meanwhile, the CMOS time runs independently of the time at the time server. When the W32Time service stops, either because the system shuts down or the service is stopped, the system time is an accurate reflection of the time at the time server.

However, the W32Time service does not guarantee that the CMOS clock is updated. Therefore, the CMOS time may differ from the time of the time server and may become out of synchronization.

This behavior may cause a time synchronization problem for the system clock after you restart the computer or after the system shuts down unexpectedly.

After you apply this hotfix, the W32Time service synchronizes the current system clock to the CMOS clock when the W32Time service shuts down. This guarantees that the time remains synchronized after the restart.

Please download the hotfix from Microsoft directly:

- http://support.microsoft.com/kb/951531

7.3 Memory Optimizations

7.3.1 General

There are two main parameters which can be used in order to configure the amount of working memory (RAM) which is being required by ***X-Tools***. Both are found within the **Global Options Editor** for the ***X-Tools Server***:

- Default Ring Buffer Size
- Allocation Size

In addition, also loading of offline data can require a significant amount of working memory.

7.3.2 Default Ring Buffer Size

The Default Ring Buffer Sizes specifies the amount of values which are being kept in the working memory for each online data. The amount of bytes within the working memory which are needed in order to acquire a given number of data with a given ring buffer size is roughly calculated as follows:

- AmountOfMemory [bytes] = NumberOfData [1] * RingBufferSize [Values] * SizeOfOneValue [bytes]
- SizeOfOneValue is dependent to the data type of the data and is calculated as follows:
  - SizeOfOneValue [bytes] = 8 [bytes] + n [bytes]
  - 8 bytes are needed for each timestamp
  - n specifies the number of bytes dependent to the data type of the data
    - Boolean and Integer8 values need 1 byte
    - Integer16 values need 2 bytes
    - Integer32 values need 4 bytes
    - Integer64 values need 8 bytes
    - Decimal32 values need 4 bytes
    - Decimal64 values need 8 bytes

Example:

- 100 online data of data type “Decimal32” are acquired
- Ring Buffer Size = 100000 [Values]

Calculation:

- AmountOfMemory [bytes] = 100 * 100000 * (8 + 4) [bytes] = 114 [MBytes]
- In this example, 114 MBytes are needed in order to be able to acquire the specified data with the specified ring buffer size.

Additional Notes:

- The ring buffer size of each data must be chosen so that at least 10 seconds of data can be hold. For example, in case a data is being acquired with 40 kHz, the ring buffer must contain at least 40000 * 10 = 400000 values.
  - In addition to the Default Ring Buffer Size from the **Global Options Editor** of the ***X-Tools Server***, also a more granular configuration possibility is available for applications which have to deal with data of very different speeds. Within the Device Profile Data table of each **Device Profile Editor**, the Buffer Mode and Buffer Parameters columns can be shown. With these settings the ring buffer size can be configured for each data individually (see the reference manual for more detailed information about these parameters). In addition, all output Analyzing Functions within Analyzing Models and Analyzing Scripts also allow to configure individual ring buffer sizes.
- On 32 bit operationg systems, the ***X-Tools Server*** can not allocate more than 2 or 3 GBytes of working memory. So even in case more GBytes are installed in the used machine, only 2 or 3 GBytes can be used by the ***X-Tools Server***. However, in case more GBytes of working memory are installed, other applications can use this additional memory and the full 2 or 3 GBytes may be available to the ***X-Tools Server***.
  - By default, 32 bit operating systems provide a maximum of 2 GBytes of working memory to each application. In order to increase the available working memory to 3 GBytes, follow the steps from the below links:
    - Windows XP and Server 2003:
      - see http://msdn.microsoft.com/en-us/library/windows/hardware/ff556232(v=vs.85).aspx
    - Windows 7:
      - open a command prompt with admistrator rights
      - enter “bcdedit /set IncreaseUserVa 3072” and run this command
      - restart the computer
      - see also http://technet.microsoft.com/en-us/query/ff542202
    - The ***X-Tools Server*** can benefit from the increased working memory size only in case this amount of memory is also present as physically installed RAM and it is not consumed by other applications.
- In addition to the acquiring of online data, also storing of online data via the **Storage System** and loading of offline data can need a significant amount of working memory. In case of very memory-intensive applications, these three components may have to be adjusted to each other in order to achieve best results.
- The ***X-Tools Server*** refuses to allocate additional working memory in case there is not enough physical RAM (not page file) left because the page file can cause extensive access to the hard disk and a high page file usage would slow down all other operations significantly. Thus, increasing of the page file does not make more working memory available for the ***X-Tools Server***.

7.3.3 Allocation Size

The Allocation Size specifies the size of the buffers which are being written to the hard disk during the storing. Bigger buffers require more working memory and achieve better writing performance whereas smaller buffers require less working memory but also have a significant worse writing performance. The amount of bytes within the working memory which are needed in order to store a given number of data is roughly calculated as follows:

- AmountOfMemory [bytes] = 2 [buffers] * NumberOfData [1] * AllocationSize [bytes]
- 2 buffers are needed because there is always one buffer being filled while the other one is written to the disk

Example:

- 100 online data shall be stored
- Allocation Size = 128 [kBytes]

Calculation:

- AmountOfMemory [bytes] = 2 * 100 * 128 [kBytes] = 25 [MBytes]
- In this example, 25 MBytes are needed in order to be able to store the specified data to the hard disk.

Additional Notes:

- In case the used hard disk is too slow to store all of the to-be-stored data, the **Storage System** temporarily can use more than two buffers for each to-be-stored data. In case of a temporary slow hard disk, this mechanism can ensure that the storing is continued without a data loss, as long as there is enough working memory available for the additional buffers.

7.3.4 Offline Data

The amount of bytes within the working memory which are needed in order to load a given data is roughly calculated as follows:

- AmountOfMemory [bytes] = NumberOfValues [Values] * SizeOfOneValue [bytes]
- SizeOfOneValue is dependent to the data type of the data (see above)

Example:

- data type of the offline data file = “Decimal16”
- NumberOfValues within the offline data file = 1500 [Values]

Calculation:

- AmountOfMemory [bytes] = 1500 * (8 + 2) [bytes] = 15 [kBytes]
- In this example, 15 kBytes are needed in order to be able to load the specified data.

7.3.5 Dealing with very huge Amounts of Data

For applications where a really huge amount of data must be acquired and stored, additional details must be considered:

- At the start of a Storage Profile, all of the files which shall receive the offline data must be created. In case of a few thousands of files, this operation can take several tens of seconds. The actual amount of needed time is dependent to the currently used system and especially to the speed of the hard disk.
    - The Windows Task Manager can be used in order to determine the amount of time which is needed until all files could have been created: Open the “Processes” tab and turn on the “Handles” column. After starting of a Storage Profile, one handle is being created for each offline data file - when the number of handles stays constant, the creation of files has been completed.
- In case the creation of all files takes several tens of seconds, also the ring buffers of all online data must be configured so that they can hold their values for at least the double amount of time. In case this condition is not given, the Storage Profile will suspend very fast because it will detect that it was unable to collect all of the required data.

- In case of huge amounts of data, the critical moment of the storing always is the moment of the file splitting. Thus, too little file splitting intervals would overload the system for sure and file splitting intervals below 10 minutes do not make sense.
- The Allocation Size of the **Storage System** can not simply be set to the smallest available value in case of huge amounts of data. Smaller allocation sizes lead to more write operations at the hard disk. For example, in case 20 MBytes would have to be stored per second, an allocation size of 8 kByte would lead to 20 [MBytes] / 16 [kBytes] = 1280 write operations per second. As typical access time of standard hard disks are at about 10 ms, the system would loose 1280 * 10 [ms] = 12 seconds of time only for the access time - and this is obviously too much time when there is only one second available for the whole storing. This detail must be considered in order to find the optimal setting for the Allocation Size.

Example:

- 2500 online data of type Decimal32 shall be acquired and stored
- all 2500 online data are acquired with 100 Hz
- the used computer has 4 GByte of RAM installed
- during a test (see above) it could be detected that the creation of 2500 files needs about 15 seconds at the used PC

Calculations:

- Ring Buffer Size
  - In case 15 seconds are needed in order to create all needed files, the ring buffer has to contain at least 30 seconds of data. At a data rate of 100 Hz (100 values per second), the ring buffer of each data must contain at least 100 * 30 = 3000 values.
  - AmountOfMemory [bytes] = 2500 * 3000 * (8 + 4) [bytes] = 85 [MBytes]
- Allocation Size
  - With the given framework, we expect to store 2500 [data] * (8 + 4) [bytes] * 100 (values per second) = 2,9 [MBytes] of data per second (plus some additional bytes for header information).
  - In order to be safe, we want to achieve an average of not more than 50 write operations to the hard disk per second. 2,9 [MBytes) / 50 = 59 [kBytes] - the next higher available Allocation Size is 64 kBytes.
  - AmountOfMemory [bytes] = 2 * 2500 * 64 [kBytes] = 313 [MBytes]
- Summary
  - We need at least 85 [MBytes] for the online data and 313 [MBytes] for the storing of this framework. Higher values are suggested both for the ring buffer size and for the allocation size in case it is possible in respect to the rest of the system. The higher these values are the more reserves are available for the ***X-Tools Server*** - however, too high values will consume more than the available amount of physical memory which suspends the affected Configuration Files.

7.3.6 Memory Management

The ***X-Tools Server*** constantly monitors the memory which is available within the whole system. In case the total memory usage of the system is too high, the ***X-Tools Server*** automatically starts to suspend its currently running Configuration Files in order to give memory back to the system. This is being done in order to avoid that Windows starts to swap the data buffers of the ***X-Tools Server*** to the hard disk, because if this swapping would be done the access (reading and writing) times to the data buffers could increase drastically and the behavior of the ***X-Tools Server*** could become unstable.

Only the amount of physically installed memory is relevant for the calculations of the ***X-Tools Server***, the configured size of the page file of Windows is ignored.

Suspending of Configuration Files is being performed in the following order:

- Monitoring Views are suspended first.
- Analyzing Models are suspended next.
- Analyzing Scripts are suspended next.
- Storage Profiles are suspended next.
- Device Profiles are suspended at last.

As soon as there is enough memory available in the system again, the ***X-Tools Server*** starts to resume its Configuration Files again. Resuming is being performed in the following order:

- Device Profiles are resumed first.
- Storage Profiles are resumed next.
- Analyzing Scripts are resumed next.
- Analyzing Models are resumed next.
- Monitoring Views are resumed at last.

In cases where the ***X-Tools Server*** suspends Configuration Files because of low memory conditions, the following measures can be performed:

- There may be other Windows applications running which consume large amounts of memory. Stop these applications in order to make the memory available for the ***X-Tools Server***.
- The amount of tasks which have to be performed by the ***X-Tools Server*** in parallel can be reduced. It may be possible to stop Configuration Files which are not needed any more.
- It may make sense to reduce the size of the buffers for online data. The default value is found within the **Global Options Editor**, additional configurations are possible within each **Device Profile Editor** and via the input arguments of all output Analyzing Functions.
- There may be a huge amount of offline data being loaded at the moment. It may be possible to remove the offline data which is not needed any more from the **MDS Explorer**.
- In case the used operating system is 32 bit and provides less than 4 GByte of physical working memory (RAM) or in case the used operating system is 64 bit, additional working memory can be installed.

7.4 Measurement Tasks

7.4.1 Frequency Analysis

7.4.1.1 General

This chapter shows how to use the **Analyzing System** in order to analyze data via a frequency analysis.

For the calculation of a frequency spectrum the Fast Fourier Transform can be used, which is an algorithm to transform a signal from the time domain into the frequency domain.

7.4.1.2 Analyzing Model

The following screenshot shows an example of an Analyzing Model for the calculation of a frequency spectrum:

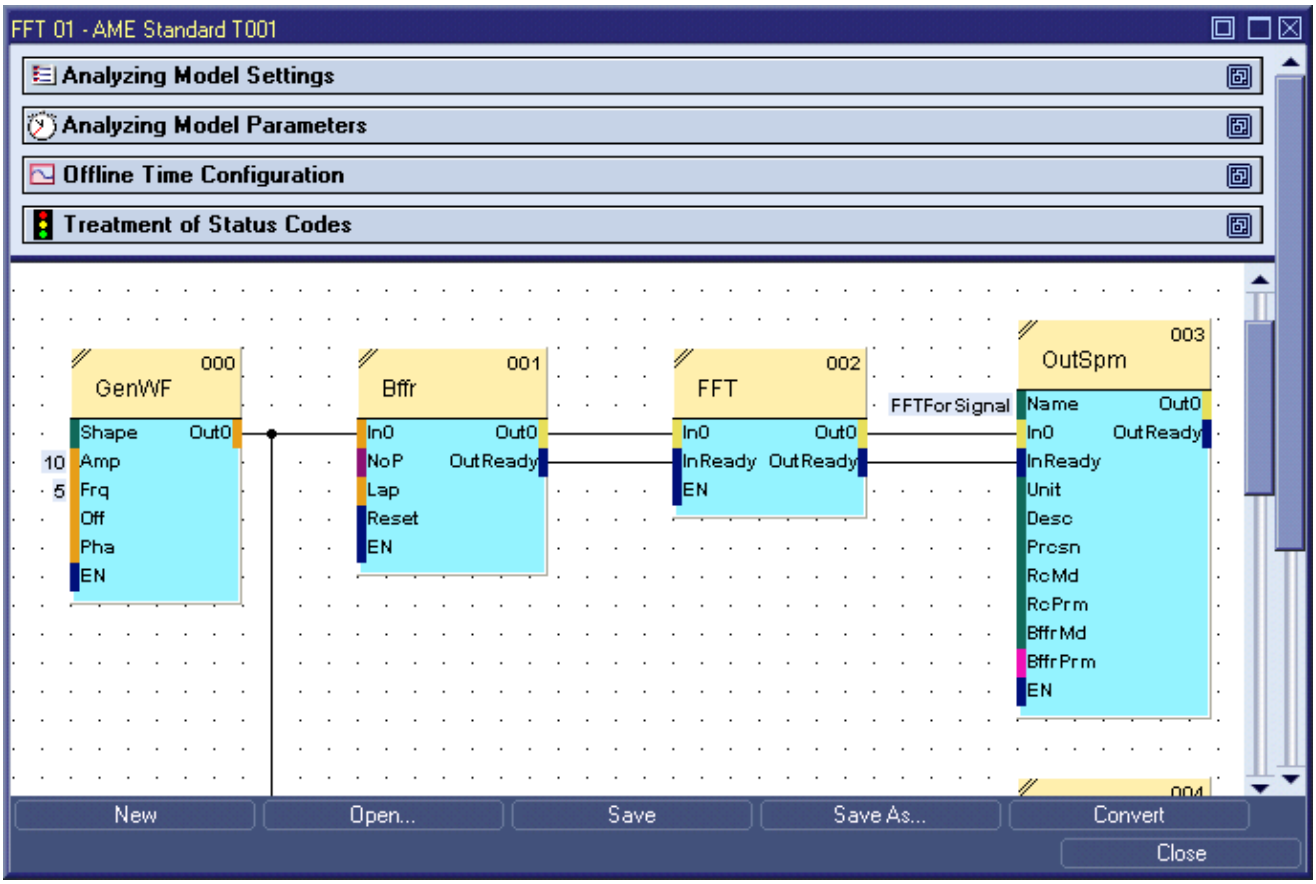


Figure 79: Example of an Analyzing Model for a Frequency Analysis

In()

- reads the signal from the system and provides it to the following Analyzing Functions

Bffr()

- creates an 1-dimensional buffer with defined size (NoP (Number of Points) must be 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768 or 65536)

FFT()

- transforms the 1-dimensional buffer from the time domain into the frequency domain

OutSpm()

- writes the calculated spectrum data to the system

7.4.1.3 Example

A concrete example is provided in order to show a real-life application. The hereby used Analyzing Model File (similar to the above screenshot, see “FFT 01”) can be found within the example location which is being delivered together with ***X-Tools***.

Input Signal

The input signal in this example is a simple sine wave with a frequency of 5 Hz and an amplitude of 10:

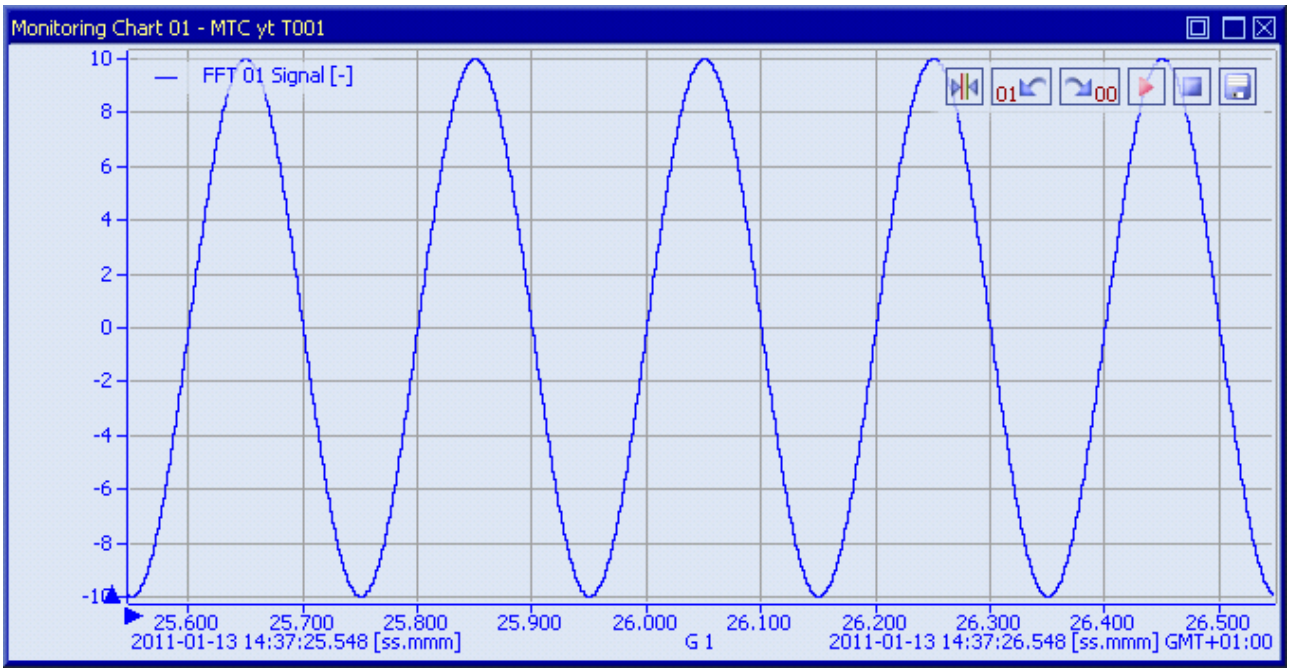


Figure 80: Example of the Input Signal for the Frequency Analysis

Spectrum

The following screenshot shows the calculated frequency spectrum of the input signal. The spectrum displays the expected peak at the frequency of 5 Hz with the amplitude of 10:

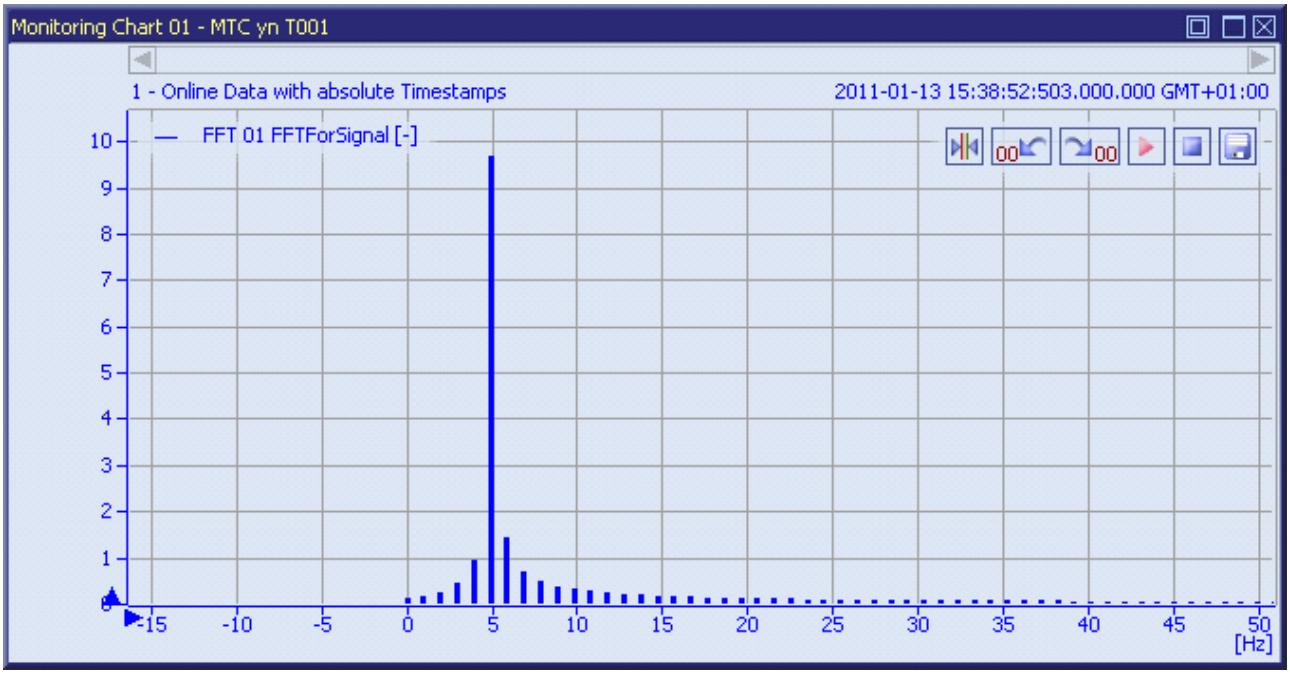


Figure 81: Example of a Frequency Spectrum which has been calculated by the Frequency Analysis

7.4.2 Spectrum Monitoring

7.4.2.1 General

This chapter shows how to automate the monitoring of a frequency spectrum with a defined alarm-spectrum.

In case certain frequencies have to be monitored (e.g. gear tooth frequencies) it is possible to set alarm levels at defined frequencies.

7.4.2.2 Analyzing Model

The following screenshot shows an example of an Analyzing Model which monitors a frequency spectrum:

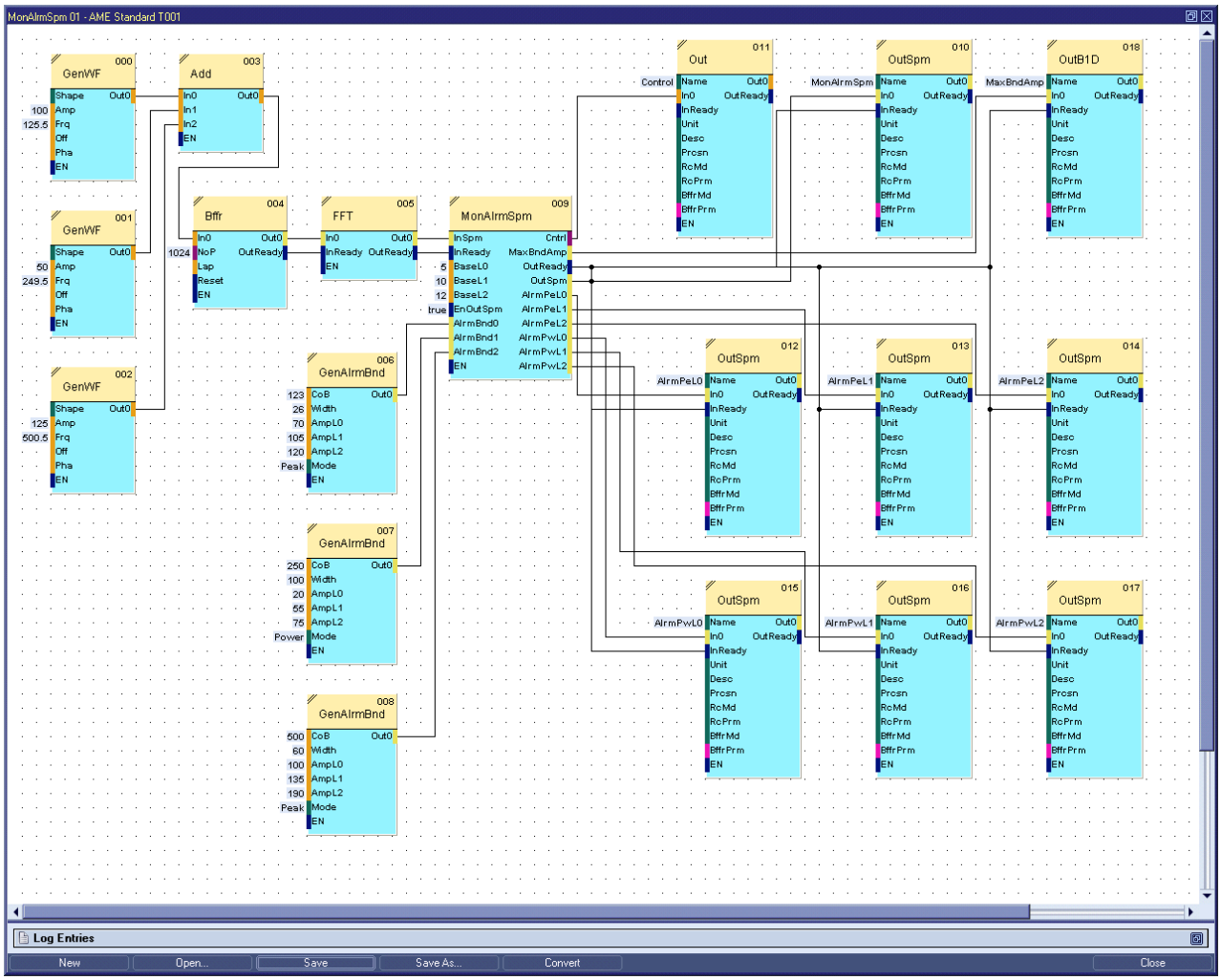


Figure 82: Example of an Analyzing Model which monitors a Frequency Spectrum

In()

- reads the signal from the system and provides it to the following Analyzing Functions

Bffr()

- creates an 1-dimensional buffer with defined size (NoP must be 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768 or 65536)

FFT()

- transforms the 1-dimensional buffer from the time domain into the frequency domain

GenAlrmBnd()

- generates an alarm band
- each alarm band is defined by its centre, width and the amplitudes of the alarm levels
- at the argument “Mode” it can be chosen, if the alarm band is monitoring “peak” values or signal “power” values

MonAlrmSpm()

- generates alarm spectra (defined by the connected alarm bands and the base levels) and monitors the connected spectrum
- output argument “Cntrl” gets the current monitoring status and “MaxBndAmp” gets the maximal peaks of the base band and each band
- for a better comparison between alarm spectra (in case of signal “power” bands) a derived spectrum can be output via “OutSpm”
- there are additional output arguments for the visualization of alarm spectra (peak or power)

Out()

- writes signals of simple data types (Booleans, Integers or decimals) to the system so that it can be used by the other systems of ***X-Tools***

OutB1D()

- writes data of data type “1-dimensional buffer” to the system so that it can be used by the other systems of ***X-Tools***

OutSpm()

- writes data of data type “spectrum” to the system so that it can be used by the other systems of ***X-Tools***

7.4.2.3 Example

A concrete example is provided in order to show a real-life application. The hereby used Analyzing Model File (similar to the above screenshot, see “MonAlrmSpm 01”) can be found within the example location which is being delivered together with ***X-Tools***.

Input Spectrum

In this example the input spectrum consists out of three added sine waves at different frequencies and with different amplitudes (see screenshot below):

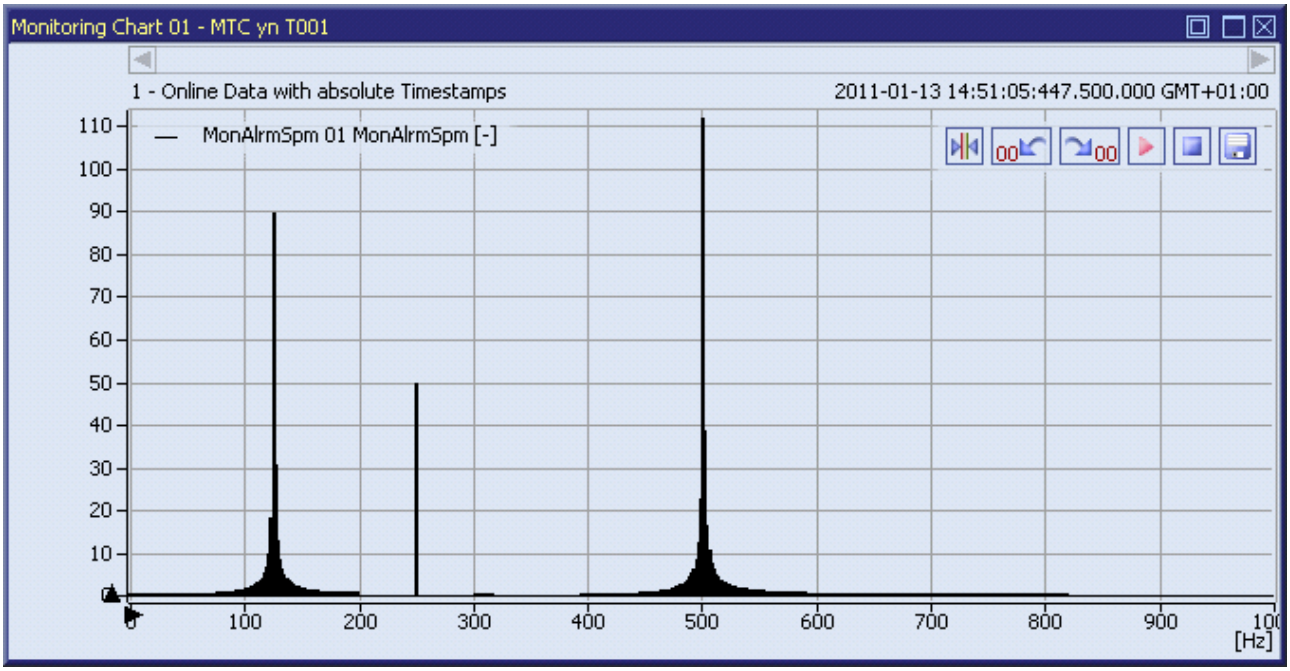


Figure 83: Example of the Input Spectrum for the Spectrum Monitoring

Alarm Spectrum

The following screenshot shows the input spectrum together with the alarm spectra. Within the given example the second peak exceeds the first alarm level of the second band:

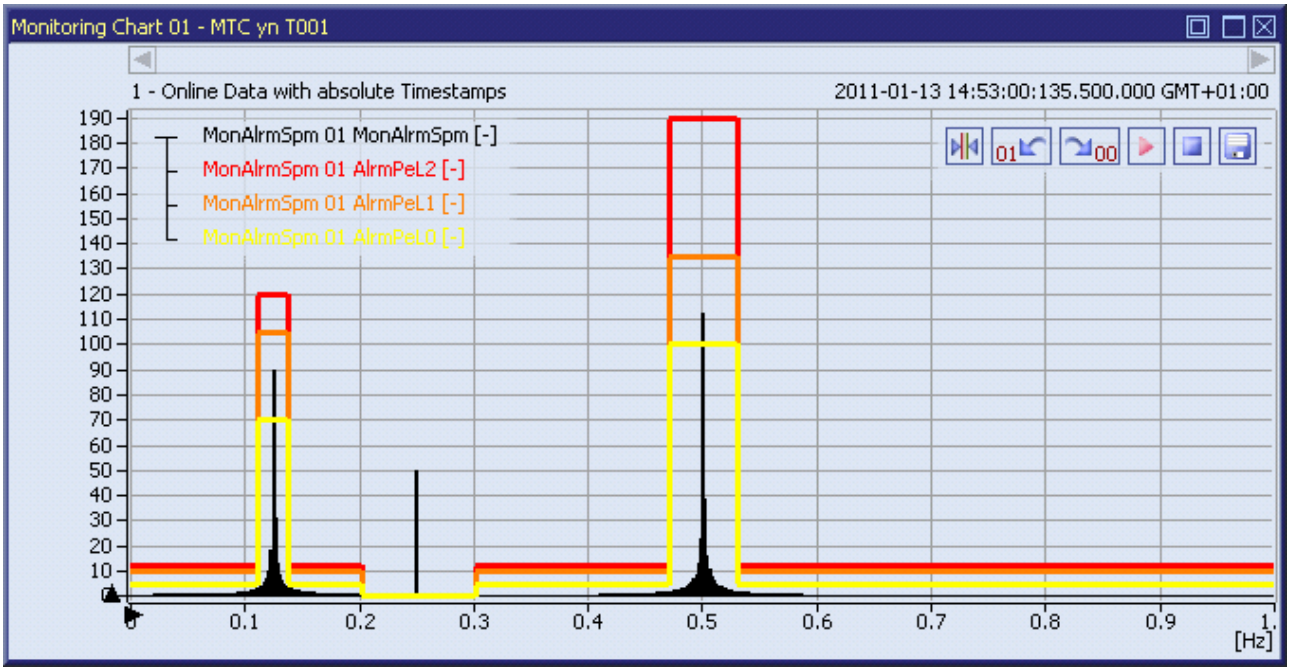


Figure 84: Example of the Input Spectrum with Alarm Spectra

Alarm Detection

There are two ways the monitored alarm can be detected.

The first way is an entry within the System Log:

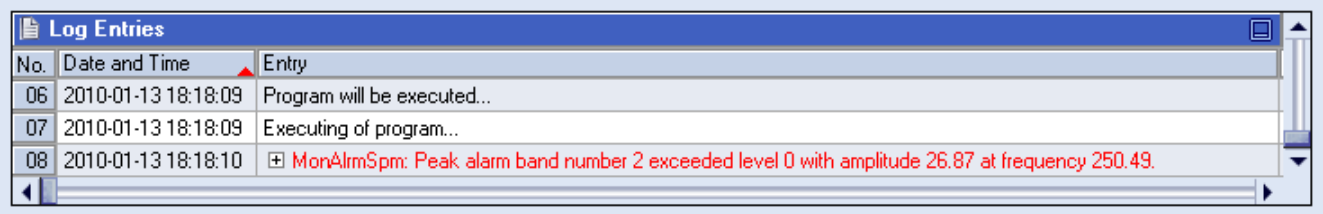


Figure 85: Example of an Alarm Log Entry

The second way is to detect it via the control signal from the output argument “Cntrl”. This signal is an integer value between 0 and 3, where

- 0 means “no alarm”
- 1 means “alarm level 0 exceeded”
- 2 means “alarm level 1 exceeded”
- 3 means “alarm level 2 exceeded”

Band Maxima

As mentioned before, there is also a way to get the maxima of each band, e.g. for trend analysis. Every time a spectrum is being monitored the according maxima (base band and each alarm band) are written to the 1-dimensional buffer “MaxBndAmp”.

It can be chosen either to use this 1-dimensional buffer or to use single values via the Analyzing Function “Get-BffrM”. Refer to the reference manual of the “Analyzing System”, chapter “GetBffrM() - GetBufferMember()” for detailed information about this Analyzing Function.

Derived Spectrum

In case there is a “power” alarm band used together with “peak” alarm bands, it is often hard to compare the input spectrum and the alarm spectra.

Therefore a derived input spectrum can be used. The derived spectrum is a copy of the input spectrum with the difference at the frequency range of each “power” band.

At these ranges the derived input spectrum doesn’t display the peak values, but the calculated signal “power” value in the middle of the frequency range. See the following screenshot, where the second alarm band was changed to a “power” band and the derived input spectrum is used in order to compare it more easily to the “power” alarm spectrum:

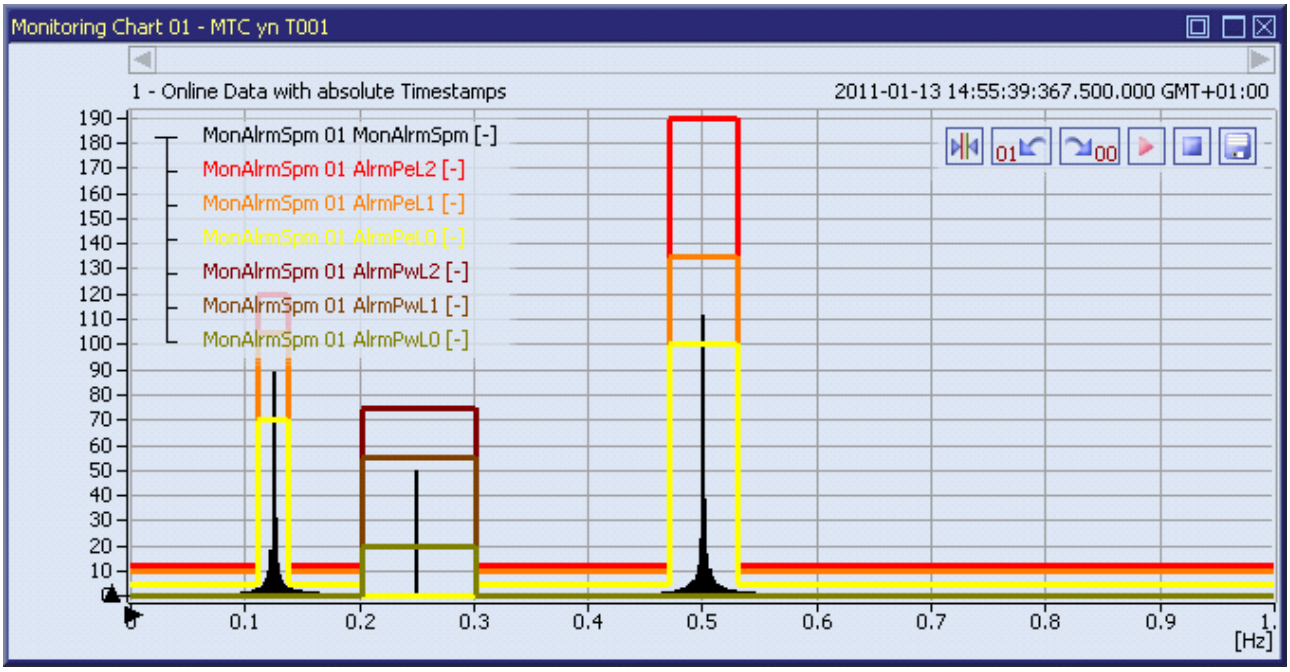


Figure 86: Example of the derived Input Spectrum with Alarm Spectra

7.4.3 Order Analysis

7.4.3.1 General

This chapter shows how to use the **Analyzing System** in order to analyze data via an order analysis. The order analysis is used mainly in order to monitor rotating machinery with variable speed.

For the calculation of an order spectrum the vibration signal has to be changed from time equidistant to angle equidistant. Therefore it is necessary to measure the speed (via impulses of a trigger) additionally to the vibration. In an additional step the vibration signal has to be re-sampled depending to the speed.

After this order re-sampling the normal FFT algorithm can be applied.

7.4.3.2 Analyzing Model

The main part for the order analysis is done by the Analyzing Function “InputOrderResampling”. Within this Analyzing Function the vibration and trigger signal is defined together with some other configuration parameters.

The following screenshot shows an example of an Analyzing Model for the calculation of an order spectrum:

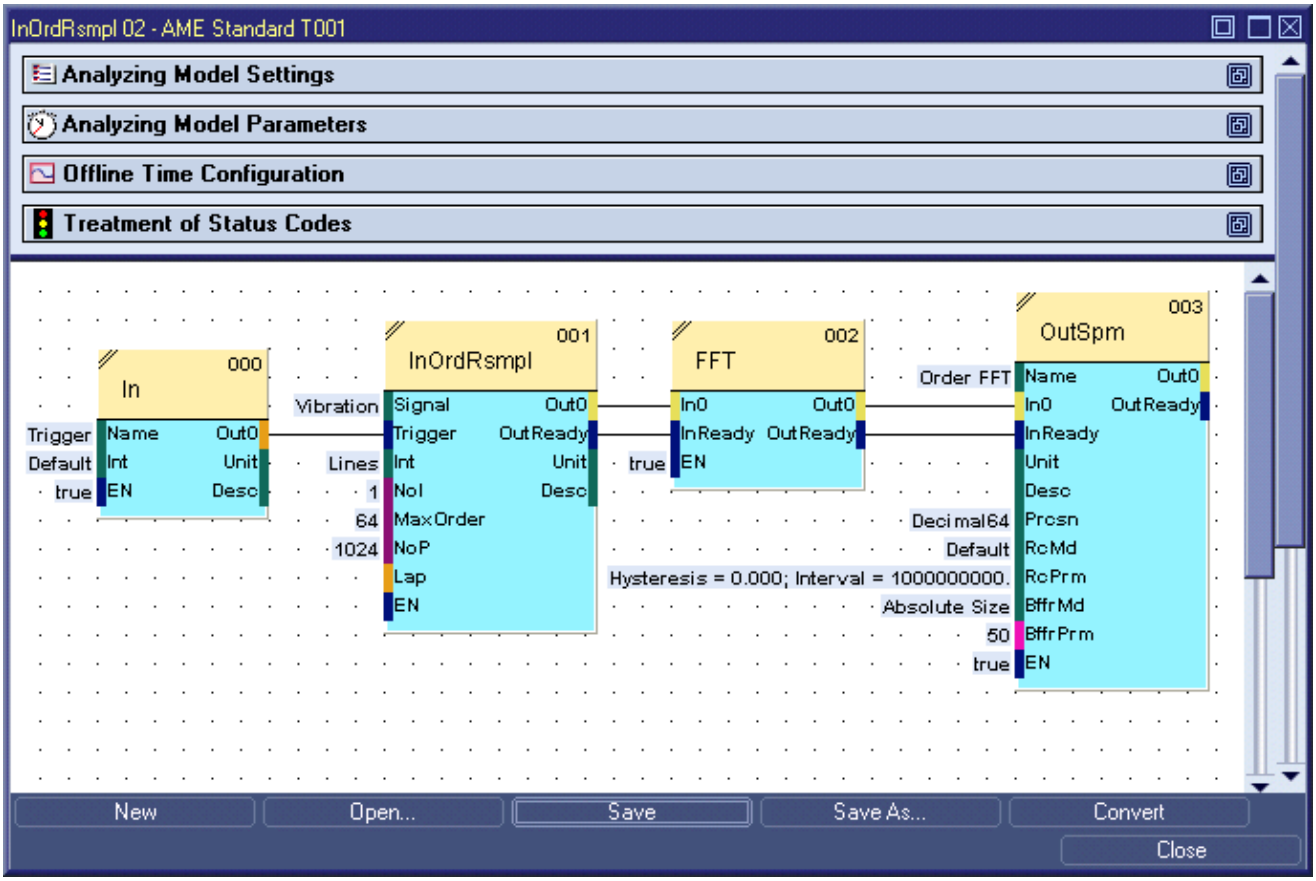


Figure 87: Example of an Analyzing Model for the Order Analysis

In()

- reads the signal from the system and provides it to the following Analyzing Functions

InOrdRsmpl()

- this Analyzing Function performs the main part of the order analysis
- the vibration signal is defined by the name which is written at argument “Signal”
- the trigger signal is connected to the “Trigger” argument
- the argument “NoI” defines how many impulses the trigger signal generates per revolution
- the argument “MaxOrder” defines the maximum order, which is displayed in the order spectrum

FFT()

- transforms the order re-sampled 1-dimensional buffer from the time domain into the order domain

OutSpm()

- writes the calculated spectrum data to the system

7.4.3.3 Example

A concrete example is provided in order to show a real-life application. The hereby used Analyzing Model File (similar to the above screenshot, see “Files\Example Location\Analyzing Model Files\Input\InOrdRsmpl 02”) can be found within the example location which is being delivered together with ***X-Tools***. In addition, also the used offline data (“Trigger” and “Vibration”) must be loaded from the offline data location which is being delivered together with ***X-Tools***.

Input Signals

There are two needed input signals, the vibration signal and the according trigger signal.

In the hereby given example the vibration signal consists out of two added sine waves. The first sine has a frequency of 10 Hz and an amplitude of 5. The second sine wave has a frequency of 100 Hz and an amplitude of 10. The according trigger signal (which represents the speed) is an impulse signal with a frequency of 10 Hz. The following screenshot shows both signals:

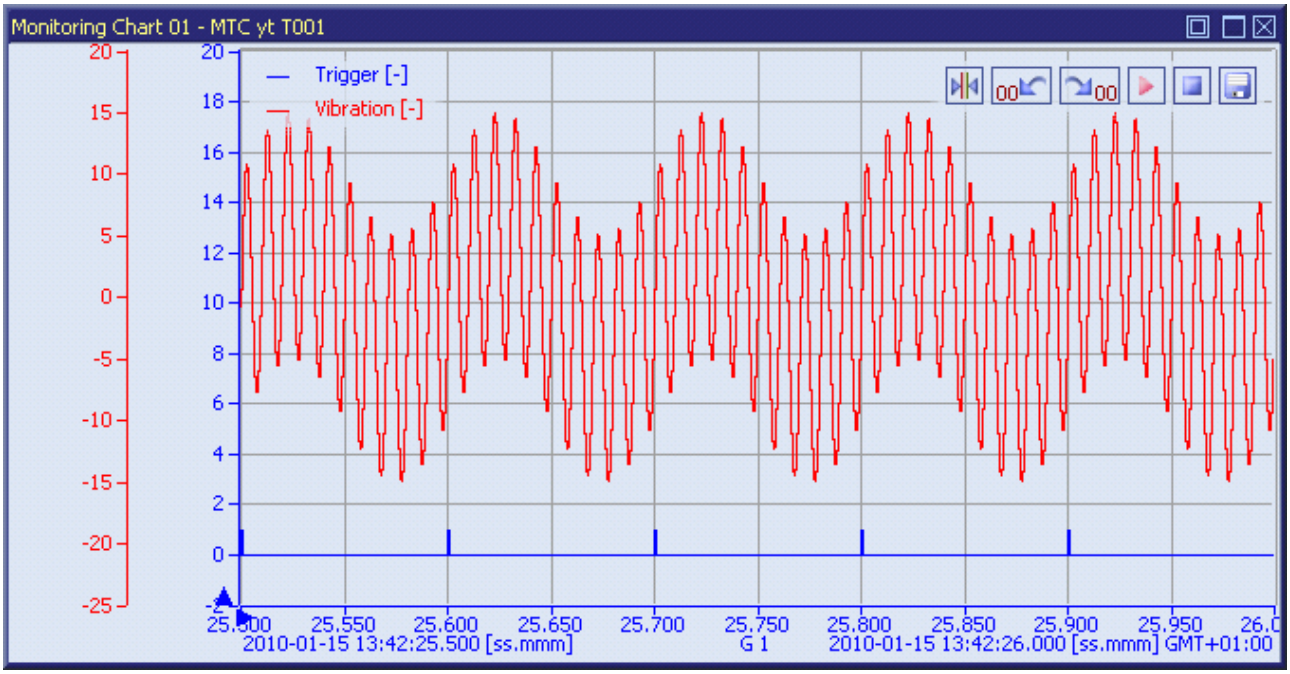


Figure 88: Example of the Input Signals for the Order Analysis

Spectrum Calculation

In order to get a better understanding of the order analysis, additionally to the order spectrum there is also a frequency spectrum calculated. The order spectrum is calculated by the Analyzing Model from above and the frequency spectrum is calculated by an Analyzing Model, which can be found within the example Configuration File location.

The following screenshot shows the two calculated spectra:

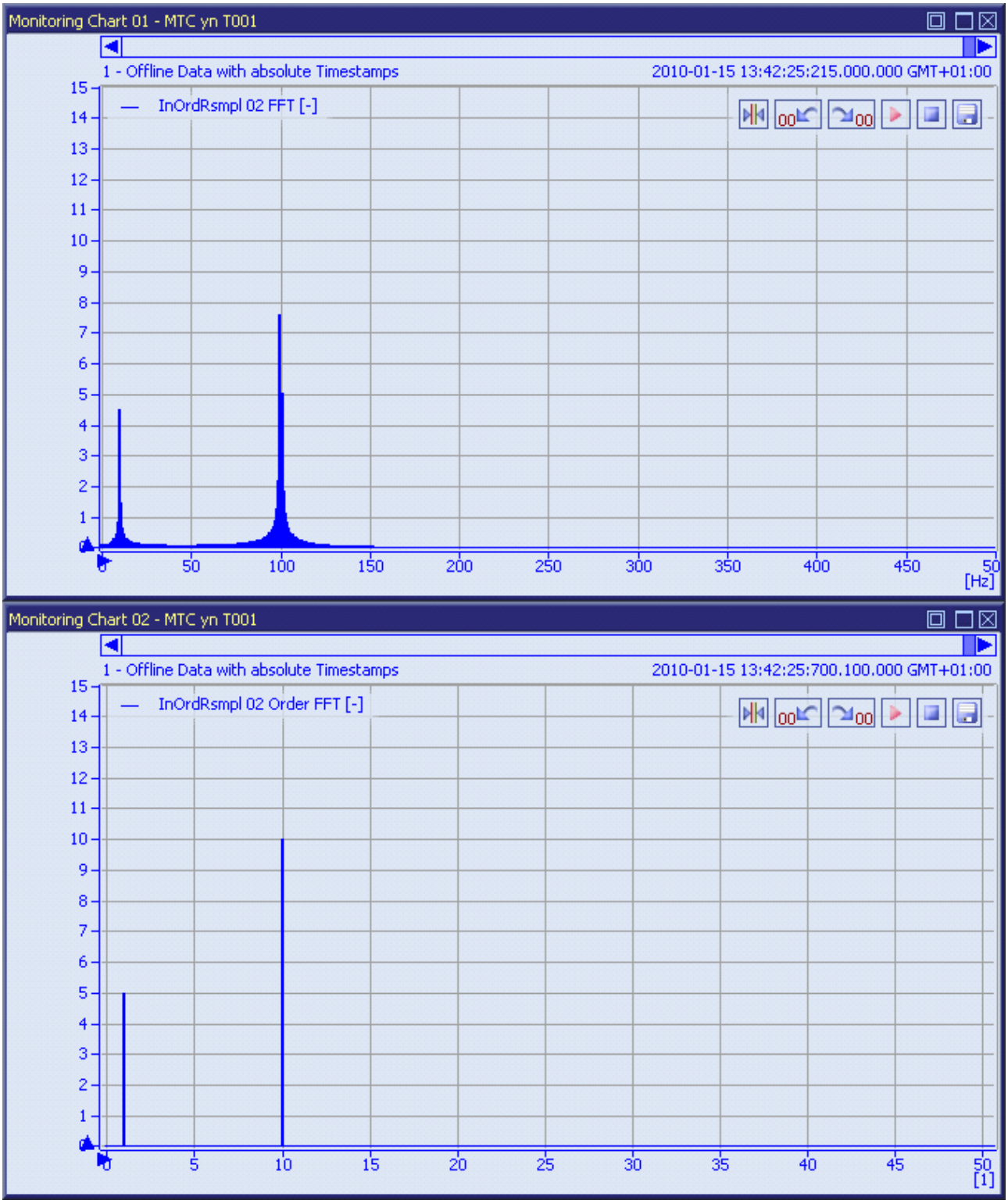

Monitoring Chart 01 - MTC yn T001
1 - Offline Data with absolute Timestamps
2010-01-15 13:42:25:215.000.000 GMT+01:00
InOrdRsmpl 02 FFT [-]
[Hz]
Monitoring Chart 02 - MTC yn T001
1 - Offline Data with absolute Timestamps
2010-01-15 13:42:25:700.100.000 GMT+01:00
InOrdRsmpl 02 Order FFT [-]
[1]

Figure 89: Example of a Frequency Spectrum and of an Order Spectrum

The frequency spectrum is displayed at the top of the figure above. The two sine waves can be seen at the according frequencies with their amplitudes. The smearing of the peaks is caused by the leakage effect.

The calculated order spectrum is shown at the lower part of the figure. The unit of the x-axis is not anymore Hertz [Hz], but orders [1]. Since the “trigger” signal was defined as an impulse signal with a constant frequency of 10 Hz, each order represents 10 Hz. According to this the order spectrum consists out of two peaks. The first peak is at order 1 with amplitude around 5 and the second peak at order 10 with amplitude around 10.

7.4.4 Sequence Control

7.4.4.1 General

This chapter shows different methods which can be used together with the sequence control.

First, it shows how to use the Analyzing Functions Start() and Stop() in order to safely start a Configuration File. “Safely” means that in case a Configuration File suspends, it will be stopped before moving on in the sequential calculation.

Second, it will be shown how to define an alarm-dependent storing in parallel to a time-dependent storing. The provided Analyzing Script stores the configured data each defined time interval and every time when an alarm has been detected.

7.4.4.2 Example

The “Sequence Control” directory of the Analyzing Scripts within the example location contains the Analyzing Script “Sequence Control 05”. This Analyzing Script shows an example for the safe start of Configuration Files and for the alarm- and time- dependent storing. Because the used input signals have a random noise, the alarm will occur not always but only sometimes. The alarm level can be changed within the Analyzing Model “Sequence Control 03” in order to receive the alarm event more or less often.

- safe start of a Configuration File:

```
var SafeStart(var ProfileName, var TimeOut)
{
    // start the Configuration File "ProfileName" -
    // if it suspends we wait only for the amount of seconds which are defined through "TimeOut"
    Start(ProfileName,'Wait',TimeOut);

    // in case the Configuration File was suspended, it will be stopped here
    Stop(ProfileName);
}
```

- main function, which defines the time interval and calls the measurement and analyzing tasks:

```
var main()
{
    // get the current time and define a flag to force storing
    var NextStore = GetTime();
    var StoreNow  = false;

    // define storing interval in seconds, here every 120s will be stored
    var StoringInterval = 120;

    // run forever
    while (1)
    {
        // check if time for the next store has passed
        if (GetTime() >= NextStore)
        {
            StoreNow  =  true;
            NextStore += StoringInterval;
        }

        // run measurement task
        PerformCalculations(StoreNow);
```

```
        // reset the storing flag
        StoreNow = false;
    }
}
```

- measurement and analyzing tasks, which start and stop the different Configuration Files:

```
var PerformCalculations(var StoreNow)
{
    // start the Interface Profile which enables the simulated devices
    Start("://Simulation T001.ipf.xts","DoNotWait");

    // start the Device Profiles of the simulated devices
    Start("://ION AnalogInput T001.dpf.xts","DoNotWait");

    // execute the desired calculations (one after the other) using SafeStart()
    SafeStart("://Sequence Control/Sequence Control 01.amf.xts",100);
    SafeStart("://Sequence Control/Sequence Control 03.amf.xts",100);

    // $'AlarmSignal' is generated by "Sequence Control 03.amf.xts" in case of an Alarm
    var Alarm = IsDataPrsnt($'AlarmSignal');

    // store only if an alarm happened or the time interval has passed
    if (Alarm || StoreNow)
    {
        // store the calculated data
        SafeStart("://Sequence Control 03.spf.xts",100);
    }

    // stop the Device Profiles
    Stop("://ION AnalogInput T001.dpf.xts");
    Stop("://ION SIMATIC S7 PN.dpf.xts");

    // stop the Interface Profile
    Stop("://Simulation T001.ipf.xts");

    // remove the calculated data
    RemoveData("://Sequence Control/Sequence Control 01.amf.xts");
    RemoveData("://Sequence Control/Sequence Control 03.amf.xts");
}
```

7.4.5 X/Y Envelopes

7.4.5.1 General

This chapter shows how to use the **Analyzing System** in order to create x/y envelopes, how to visualize the calculated envelopes and how to use them for monitoring of current online data against violation of these envelopes.

The visualizations, Analyzing Models and Analyzing Scripts from this chapter can be tried out because the input data and the Analyzing Models/Analyzing Scripts are delivered together with ***X-Tools***. The input data is found within the "..\Server\User\Example Offline Files\X-Tools Offline Data Files\Measurement Tasks\XY Envelopes" path of the ***X-Tools*** installation and must be loaded from there, whereas the used Analyzing Models, Analyzing Scripts and Monitoring Views are part of the example location which is delivered together with ***X-Tools***. The "Measurement Tasks\XY Envelopes" subdirectory of each Configuration File type contains the Configuration Files to which the following descriptions refer (see also point 7.4.5.6).

The provided input data has been created by storing of the output of the "10 Create Reference x and y Data" Analyzing Model from the example location. For the generation of online x and y data which also shows violations of the calculated envelopes, the "40 Simulate x and y Data with Errors" Analyzing Model/Analyzing Script from the example location must be started.

7.4.5.2 Creation of X/Y Envelopes

In order to create an x/y envelope, the x and y input data must be available. The following screenshots shows the visualization of an example x and y data. This example data is delivered together with ***X-Tools*** and can be loaded from the example offline files - in addition, the Monitoring View "Example for XY Envelopes" from the example location can be opened in order to receive the following visualization:

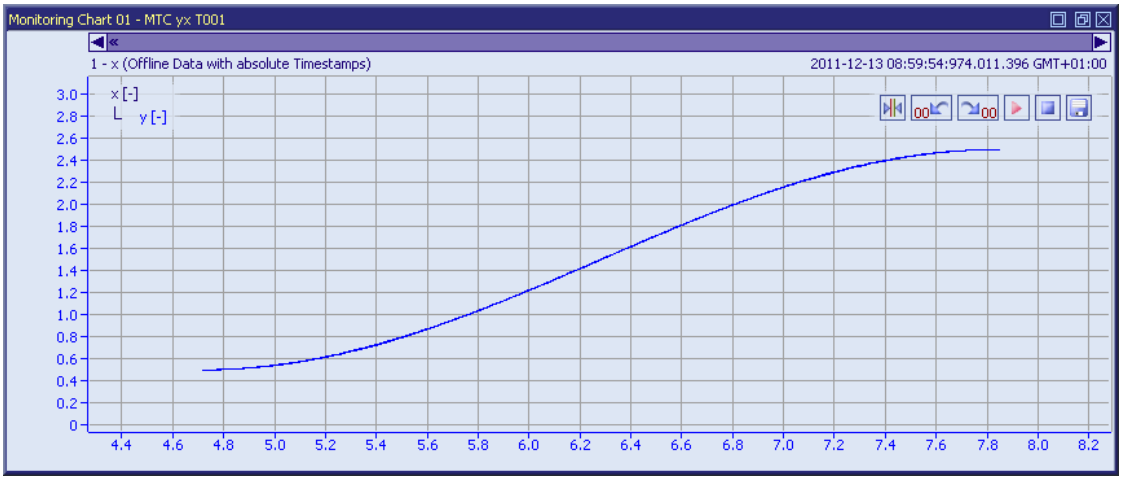


Figure 90: Example of the Input Data for an X/Y Envelope

The Analyzing Function CalculateXYEnvelope() is being used for the actual calculation. This Analyzing Function always calculates one upper and one lower band for the provided input data. In case multiple upper and lower bands shall be monitored (e.g. in case warning and an error bands are needed), CalculateXYEnvelope() must be called multiple times with different thresholds in order to calculate all needed envelopes.

CalculateXYEnvelope() must be used within a sequential Analyzing Script in order to perform the calculation. The following box shows the Analyzing Script “20 Create XY Envelope” from the example location, which shows how CalculateXYEnvelope() can be used:

```
var main()
{
    // the following offline data must be loaded:
    // - x
    // - y

    // calculate the warning band first
    $'XY Envelope StatusWarning' =
        ClcXYEnv($'x',$'y',1024,0.2,0.2,0.2,0.2,
                 "XY Envelope XBufferWarning",
                 "XY Envelope UpperWarningBand",
                 "XY Envelope LowerWarningBand");

    // calculate the error band next
    $'XY Envelope StatusError' =
        ClcXYEnv($'x',$'y',1024,0.4,0.4,0.4,0.4,
                 "XY Envelope XBufferError",
                 "XY Envelope UpperErrorBand",
                 "XY Envelope LowerErrorBand");
}
```

The created envelopes can be stored. Afterwards they can be loaded at any time in order to use them together with the Analyzing Function MonitorXYEnvelope() for the monitoring of the current x and y data against the calculated envelopes.

7.4.5.3 Visualization of X/Y Envelopes

The output of each call to CalculateXYEnvelope() are three 1-dimensional buffers which are ideal for the following monitoring through the Analyzing Function MonitorXYEnvelope(). However, in case the envelopes shall be visualized together with the original input data within a MTC yx T001, the Analyzing Function ConvertBuffer1DToTimeSeries() must be used in order to convert each 1-dimensional buffer into a time series. The following box shows the Analyzing Script “30 Prepare Visualization of XY Envelope” from the example location, which shows how ConvertBuffer1DToTimeSeries() can be used:

```
var main()
{
    // the following Analyzing Script must be started before this one:
    // - 20 Create XY Envelope

    // convert the x-buffer of the warning bands to a time series
    ConvB1DToTS("XY Envelope XBufferWarning",
                "XY Envelope XBufferWarningTS",
                "End with Input Data Time",0.029296875);

    // convert the upper warning band to a time series
    ConvB1DToTS("XY Envelope UpperWarningBand",
                "XY Envelope UpperWarningBandTS",
                "End with Input Data Time",0.029296875);

    // convert the lower warning band to a time series
    ConvB1DToTS("XY Envelope LowerWarningBand",
                "XY Envelope LowerWarningBandTS",
                "End with Input Data Time",0.029296875);

    // convert the x-buffer of the error bands to a time series
    ConvB1DToTS("XY Envelope XBufferError",
                "XY Envelope XBufferErrorTS",
                "End with Input Data Time",0.029296875);

    // convert the upper error band to a time series
    ConvB1DToTS("XY Envelope UpperErrorBand",
                "XY Envelope UpperErrorBandTS",
                "End with Input Data Time",0.029296875);

    // convert the lower error band to a time series
    ConvB1DToTS("XY Envelope LowerErrorBand",
                "XY Envelope LowerErrorBandTS",
                "End with Input Data Time",0.029296875);
}
```

For the x and y input data which is delivered together with ***X-Tools*** we know that their time interval is 30 seconds (this information can be read from the **MDS Explorer** when the data is loaded). When the time series which is output by ConvertBuffer1DToTimeSeries() shall cover exactly the same time interval (from the same start time until the same stop time) as the original x and y input data, the “TimeStampMode” parameter must be set to “End with Input Data Time” and the “TimeInterval” must be calculated with the following formula:

- TimeIntervalForConvB1DToTS = TimeIntervalOfXandY / DimensionOfClcXYEnv

For our input data and Analyzing Script this calculation delivers the following result:

- TimeIntervalForConvB1DToTS = 30 [s] / 1024 = 0.029296875

However, when the output of ConvertBuffer1DToTimeSeries() is needed only for the visualization, it does not matter whether the time intervals of the original x and y data and of the output of ConvertBuffer1DToTimeSeries() have any relation to each other - in this case, the above script can be shorter because also the default values of ConvertBuffer1DToTimeSeries() are fully OK:

```
var main()
{
    // the following Analyzing Script must be started before this one:
    // - 20 Create XY Envelope

    // convert the x-buffer of the warning bands to a time series
    ConvB1DToTS("XY Envelope XBufferWarning",
                "XY Envelope XBufferWarningTS");

    // convert the upper warning band to a time series
    ConvB1DToTS("XY Envelope UpperWarningBand",
                "XY Envelope UpperWarningBandTS");

    // convert the lower warning band to a time series
    ConvB1DToTS("XY Envelope LowerWarningBand",
                "XY Envelope LowerWarningBandTS");

    // convert the x-buffer of the error bands to a time series
    ConvB1DToTS("XY Envelope XBufferError",
                "XY Envelope XBufferErrorTS");

    // convert the upper error band to a time series
    ConvB1DToTS("XY Envelope UpperErrorBand",
                "XY Envelope UpperErrorBandTS");

    // convert the lower error band to a time series
    ConvB1DToTS("XY Envelope LowerErrorBand",
                "XY Envelope LowerErrorBandTS");
}
```

The following screenshots shows the visualization of the example x and y data together with the calculated envelopes:

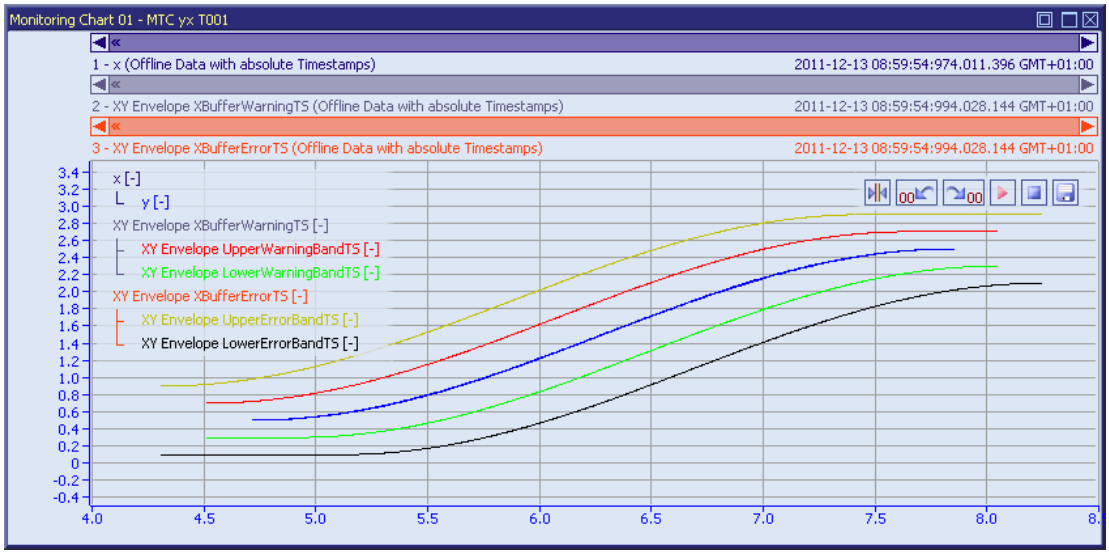


Figure 91: Example of the Input Data for an X/Y Envelope together with the calculated Envelopes

7.4.5.4 Monitoring of X/Y Envelopes

After the necessary envelopes have been calculated, they can be used for monitoring of the currently present x and y data. The following screenshot shows the Analyzing Model “50 Monitor XY Envelope” from the example location which uses the above envelopes in order to monitor the current x and y online data:

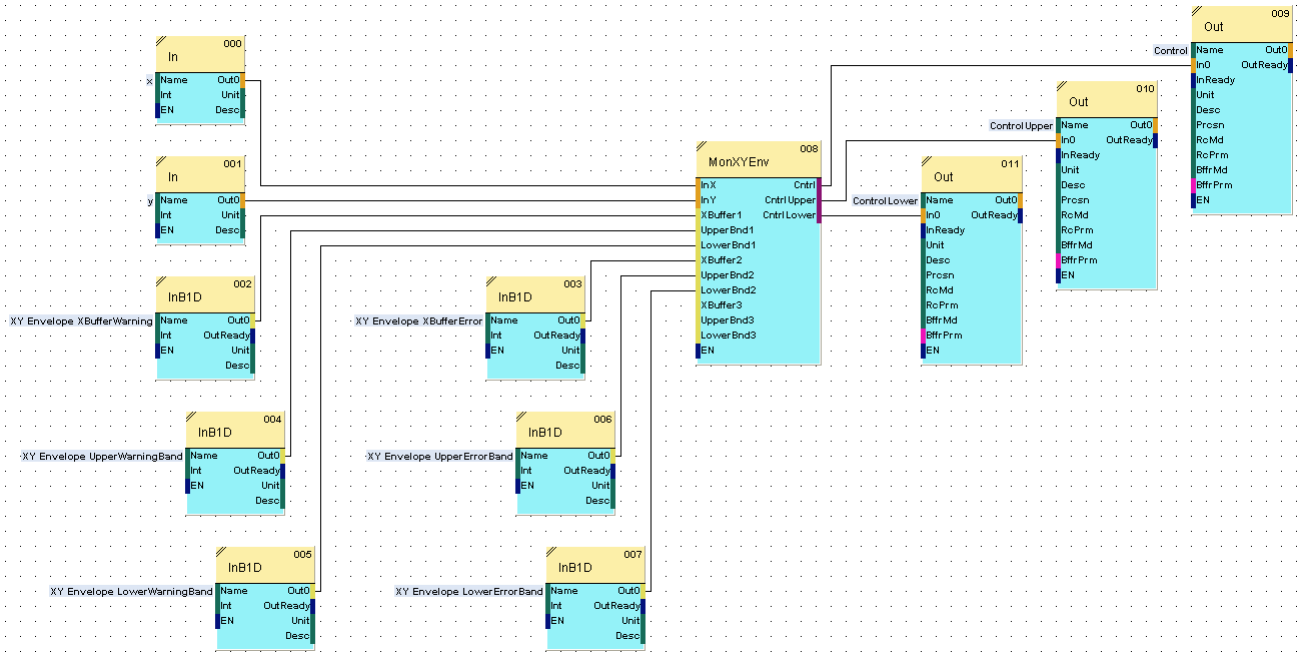


Figure 92: Example of an Analyzing Model for the monitoring of the current x and y online Data

The same task can be solved via an Analyzing Script also - the following box shows the Analyzing Script “50 Monitor XY Envelope” from the example location:

```
var main()
{
    // the following Analyzing Scripts must be started before this one:
    // - 20 Create XY Envelope
    // - 40 Simulate x and y Data with Errors

    ~'Control' =
        MonXYEnv(~'x',~'y',
                 $'XY Envelope XBufferWarning',
                 $'XY Envelope UpperWarningBand',$'XY Envelope LowerWarningBand',
                 $'XY Envelope XBufferError',
                 $'XY Envelope UpperErrorBand',$'XY Envelope LowerErrorBand',
                 0,0,0,true,
                 ~'ControlUpper',~'ControlLower');
}
```

Figure 93: Example of an Analyzing Script for the monitoring of the current x and y online Data

Whenever a violation of the specified envelopes is detected, **MonXYEnv()** creates an according log entry.

7.4.5.5 Example

The offline data, Analyzing Models and Analyzing Scripts from the example locations which are delivered together with ***X-Tools*** can be used in order to reproduce the above steps (see also point 7.4.5.1).

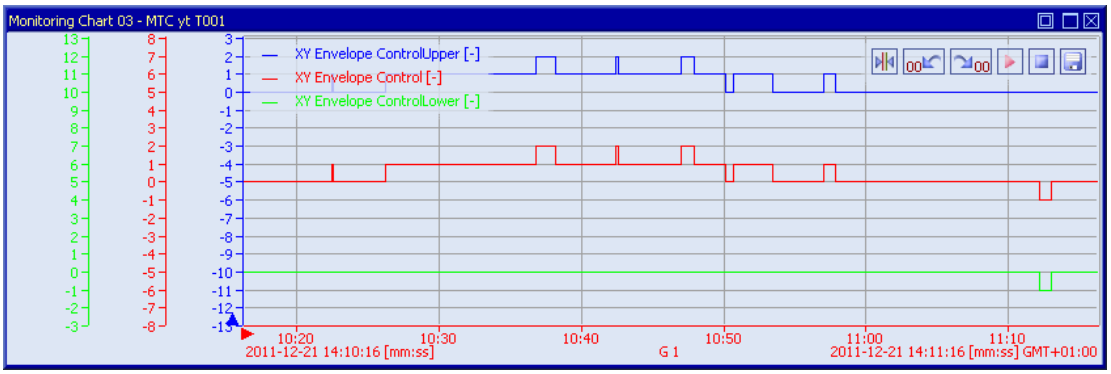


Figure 94: Example of the Output of MonitorXYEnvelope()

7.4.5.6 Summary

The following Configuration Files are being used by this chapter:

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\XY Envelopes")

Name	Description
10 Create Reference x and y Data	This Analyzing Model can be used in order to create the reference x and y data (without errors). It is typically not needed because the data from this Analyzing Model is available as example offline data.
40 Simulate x and y Data with Errors	This Analyzing Model simulates online x and y data with small errors so that violations of the calculated envelopes take place.
50 Monitor XY Envelope	This Analyzing Model monitors the online x and y data against the created x/y envelopes.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\XY Envelopes")

Name	Description
20 Create XY Envelope	This Analyzing Script creates all x/y envelopes out of the previously loaded x and y offline data.
30 Prepare Visualization of XY Envelope	This Analyzing Script converts the created x/y envelopes from a 1-dimensional buffer into a time series. This step is needed only in case the created x/y envelopes shall be monitored via the **Monitoring System**.
40 Simulate x and y Data with Errors	This Analyzing Script simulates online x and y data with small errors so that violations of the calculated envelopes take place.
50 Monitor XY Envelope	This Analyzing Script monitors the online x and y data against the created x/y envelopes.

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\XY Envelopes")

Name	Description
Visualize XY Envelopes	This Monitoring View shows the original x and y offline data, the calculated x/y envelopes, the current x and y online data and the three outputs of MonXYEnv().

Offline Data
(found in "..\Example Offline Files\X-Tools Offline Data Files\Measurement Tasks\XY Envelopes")

Name	Description
x	The x data which has been generated via the Analyzing Model "10 Create Reference x and y Data".
y	The y data which has been generated via the Analyzing Model "10 Create Reference x and y Data".

7.4.6 Basic Spectrum Envelopes

7.4.6.1 General

This chapter shows how to use the **Analyzing System** in order to create basic spectrum envelopes, how to visualize the calculated envelopes and how to use them for monitoring of current online data against violations of these envelopes.

The Monitoring Views, Analyzing Models and Analyzing Scripts from this chapter can be tried out because the input data and the Monitoring Views, Analyzing Models and Analyzing Scripts are delivered together with ***X-Tools***. The input data is found within the "..\Server\User\Example Offline Files\X-Tools Offline Data Files\Measurement Tasks\Basic Spectrum Envelopes" path of the ***X-Tools*** installation and must be loaded from there, whereas the used Analyzing Models, Analyzing Scripts and Monitoring Views are part of the example location which is delivered together with ***X-Tools***. The "Measurement Tasks\Basic Spectrum Envelopes" subdirectory of each Configuration File type contains the Configuration Files to which the following descriptions refer (see also point 7.4.6.6).

The provided input data has been created by storing of the output of the "10 Create Reference Spectrum" Analyzing Model from the example location. For the generation of an online spectrum which also shows violations of the calculated envelopes, the "30 Simulate Spectrum with Errors" Analyzing Model/Analyzing Script from the example location must be started.

7.4.6.2 Creation of Spectrum Envelopes

In order to create a spectrum envelope, the input spectrum must be available. The following screenshot shows the visualization of an example spectrum. This example data is delivered together with ***X-Tools*** and can be loaded from the example offline files - in addition, the Monitoring View "Example for Spectrum Envelopes" from the example location can be opened in order to receive the following visualization:

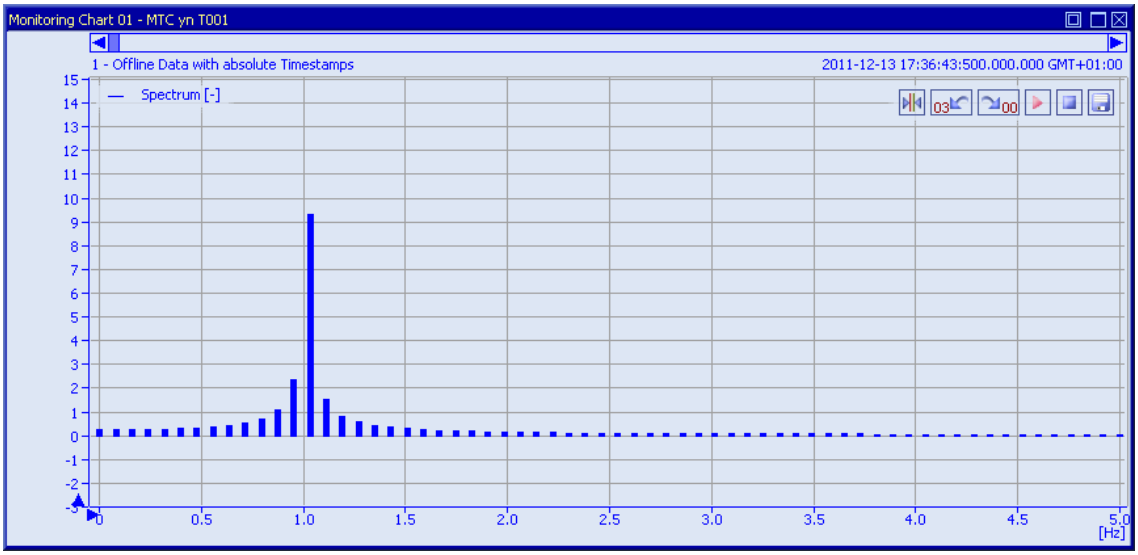


Figure 95: Example of the Input Data for a basic Spectrum Envelope

The Analyzing Function SpectrumAddition() can be used in order to add an absolute or a relative value to each amplitude of the spectrum. In case multiple envelopes shall be monitored (e.g. warning and error bands above and below the reference spectrum), SpectrumAddition() must be called multiple times with different thresholds in order to calculate all needed envelopes.

The following screenshot shows the Analyzing Model “20 Create Spectrum Envelope” from the example location which creates four envelopes out of the reference spectrum:

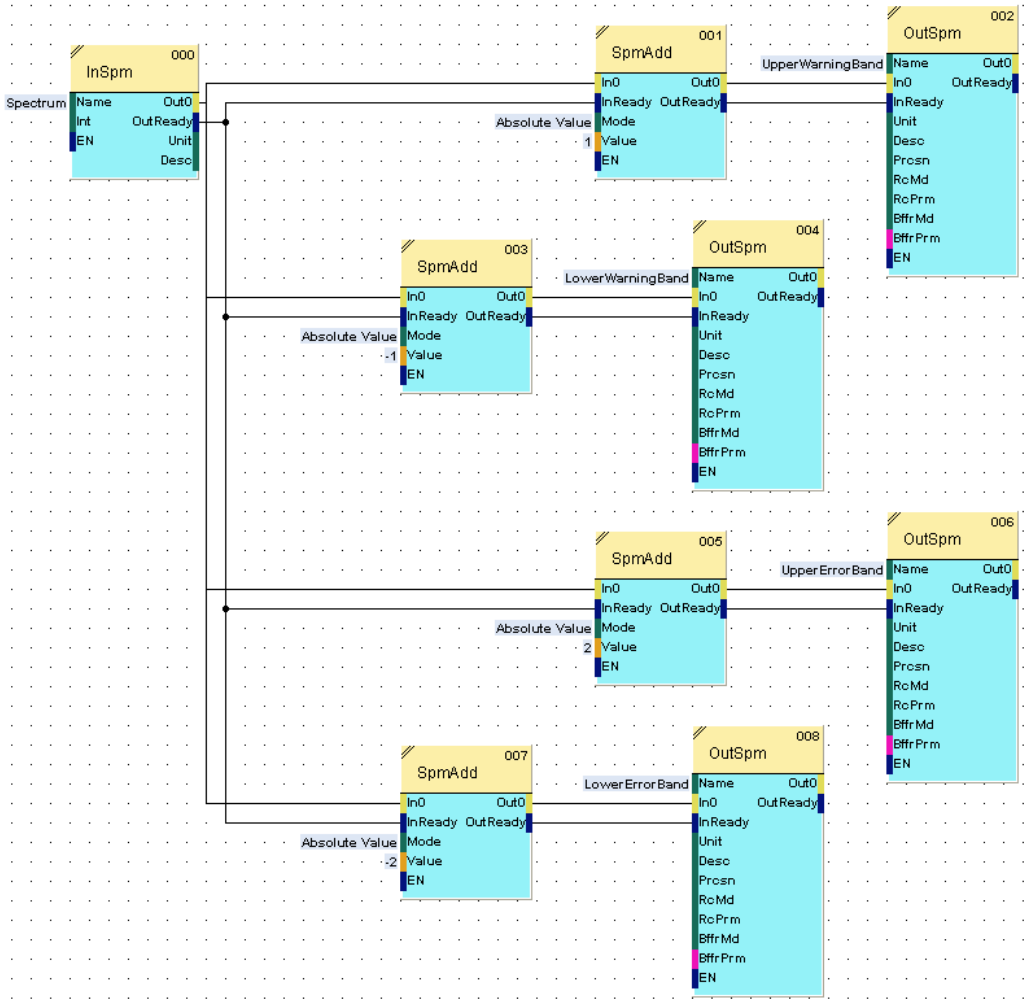


Figure 96: Example of an Analyzing Model for the Creation of Spectrum Envelopes

The same task can be solved via an Analyzing Script also - the following box shows the Analyzing Script “20 Create Spectrum Envelope” from the example location:

```
var main()
{
    // the following offline data must be loaded:
    // - Spectrum

    var Spectrum;
    var SpectrumReady;

    Spectrum = InSpm($'Spectrum',"Default",true,SpectrumReady);
    $'UpperWarningBand' = SpmAdd(Spectrum,SpectrumReady,"Absolute Value",1);
    $'LowerWarningBand' = SpmAdd(Spectrum,SpectrumReady,"Absolute Value",-1);
    $'UpperErrorBand'   = SpmAdd(Spectrum,SpectrumReady,"Absolute Value",2);
    $'LowerErrorBand'   = SpmAdd(Spectrum,SpectrumReady,"Absolute Value",-2);
}
```

Figure 97: Example of an Analyzing Script for the Creation of Spectrum Envelopes

The created envelopes can be stored. Afterwards they can be loaded at any time in order to use them together with the Analyzing Function MonitorSpectrumEnvelope() for the monitoring of the current spectrum against the calculated envelopes.

7.4.6.3 Visualization of Spectrum Envelopes

The output of SpectrumAdd() is a spectrum again and can be used directly for the visualization within the MTC yn T001. The following screenshots shows the visualization of the example spectrum together with the calculated envelopes:

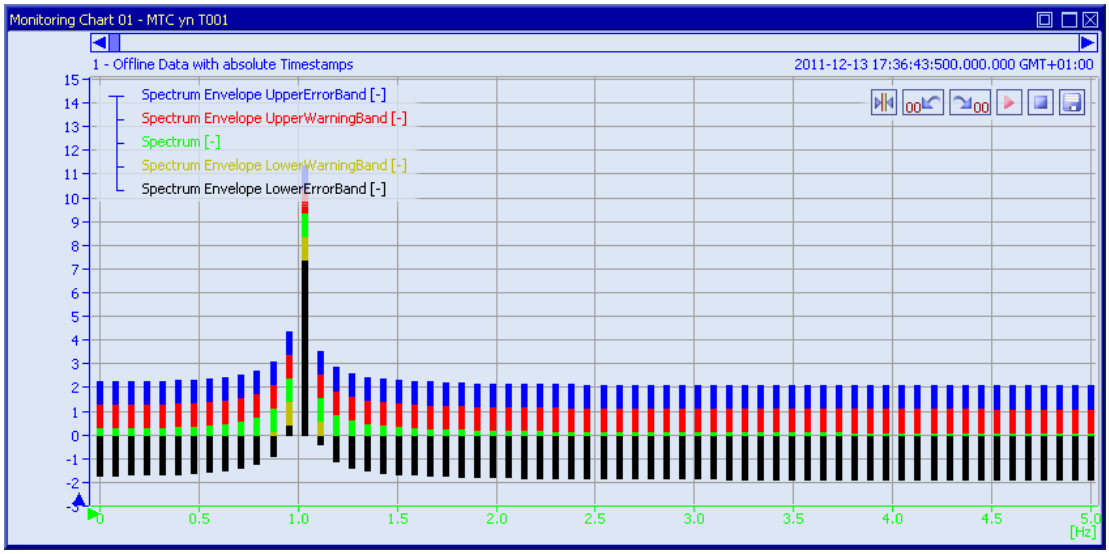

Monitoring Chart 01 - MTC yn T001
1 - Offline Data with absolute Timestamps
2011-12-13 17:36:43:500.000.000 GMT+01:00
Spectrum Envelope UpperErrorBand [-]
Spectrum Envelope UpperWarningBand [-]
Spectrum [-]
Spectrum Envelope LowerWarningBand [-]
Spectrum Envelope LowerErrorBand [-]
[Hz]

Figure 98: Example of the Input Data for a Spectrum Envelope together with the calculated Envelopes

7.4.6.4 Monitoring of Spectrum Envelopes

After the necessary envelopes have been calculated, they can be used for monitoring of the currently present spectrum data. The following screenshot shows the Analyzing Model “40 Monitor Spectrum Envelope” from the example location which uses the above envelopes in order to monitor the current online spectrum:

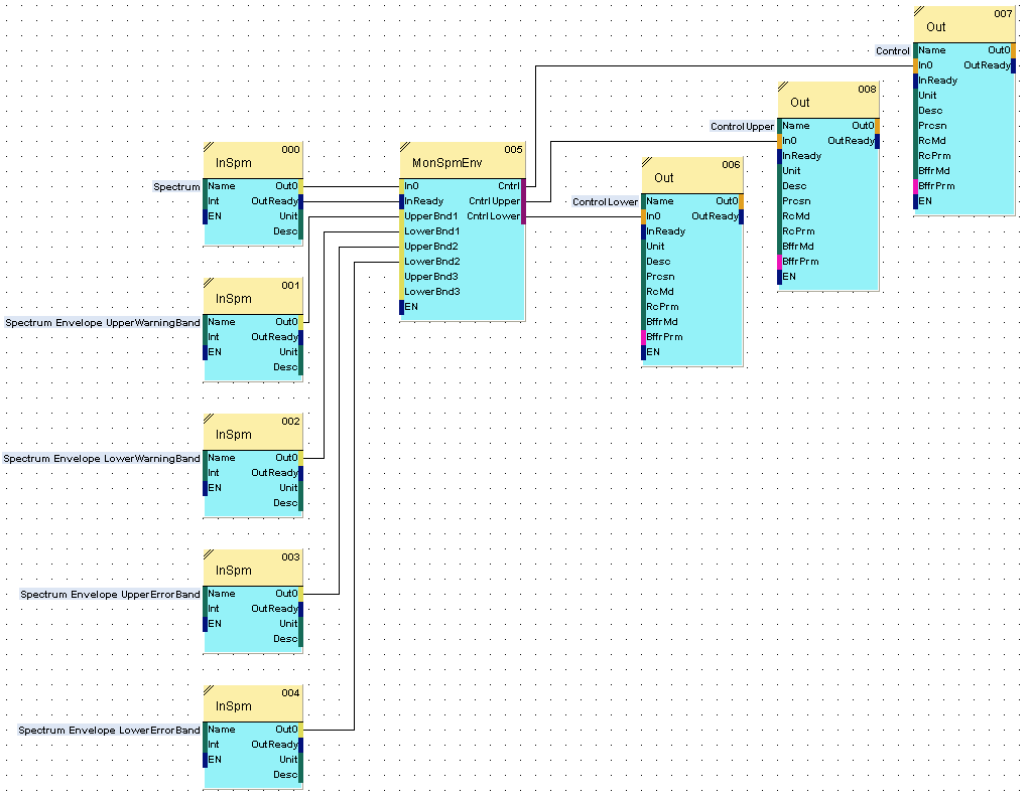


Figure 99: Example of an Analyzing Model for the monitoring of the current online Spectrum

In order to simulate some spectrum with errors, the Analyzing Model “30 Simulate Spectrum with Errors” must be started. This Analyzing Model needs about 15 seconds before it can provide the output data “Spectrum”,

thus the Analyzing Model “40 Monitor Spectrum Envelope” shall not be started before this time (= before the online data “Spectrum” appears within the **MDS Explorer**). In case the Analyzing Model “40 Monitor Spectrum Envelope” is started before the online data “Spectrum” becomes available, it may use the offline data “Spectrum” as input automatically.

The same task can be solved via an Analyzing Script also - the following box shows the Analyzing Script “40 Monitor Spectrum Envelope” from the example location:

```
var main()
{
    // the following Analyzing Scripts must be started before this one:
    // - 20 Create Spectrum Envelope
    // - 30 Simulate Spectrum with Errors

    var Spectrum;
    var SpectrumReady;

    Spectrum = InSpm(~'Spectrum',"Default",true,SpectrumReady);
    ~'Control' =
        MonSpmEnv(Spectrum,SpectrumReady,
                  $'Spectrum Envelope UpperWarningBand',$'Spectrum Envelope LowerWarningBand',
                  $'Spectrum Envelope UpperErrorBand',$'Spectrum Envelope LowerErrorBand',
                  0,0,true,
                  ~'ControlUpper',~'ControlLower');
}
```

Figure 100: Example of an Analyzing Script for the monitoring of the current online Spectrum

The Analyzing Script “40 Monitor Spectrum Envelope” explicitelly accesses the online data “Spectrum”, thus it can be started at any point in time and won’t start to perform calculations unless the simulated online data “Spectrum” appears within the **MDS Explorer**.

Whenever a violation of the specified envelopes is detected, **MonSpmEnv()** creates an according log entry.

7.4.6.5 Example

The offline data, Analyzing Models and Analyzing Scripts from the example locations which are delivered together with ***X-Tools*** can be used in order to reproduce the above steps (see also point 7.4.6.1).

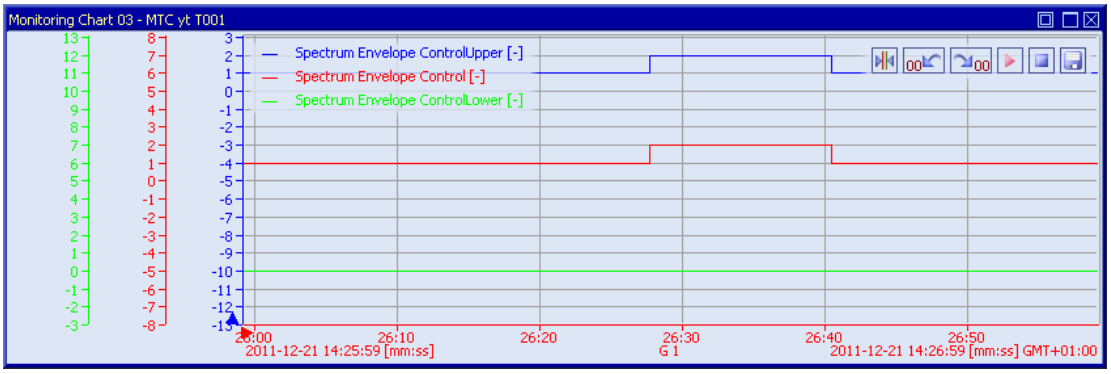


Figure 101: Example of the Output of MonitorSpectrumEnvelope()

7.4.6.6 Summary

The following Configuration Files are being used by this chapter:

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Basic Spectrum Envelopes")

Name	Description
10 Create Reference Spectrum	This Analyzing Model can be used in order to create the reference spectrum (without errors). It is typically not needed because the data from this Analyzing Model is available as example offline data.
20 Create Spectrum Envelope	This Analyzing Model creates all spectrum envelopes ouf of the previously loaded spectrum data.
30 Simulate Spectrum with Errors	This Analyzing Model simulates an online spectrum with small errors so that violations of the calculated envelopes take place.
40 Monitor Spectrum Envelope	This Analyzing Model monitors the online spectrum against the created spectrum envelopes.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Basic Spectrum Envelopes")

Name	Description
20 Create Spectrum Envelope	This Analyzing Script creates all spectrum envelopes ouf of the previously loaded spectrum data.
30 Simulate Spectrum with Errors	This Analyzing Script simulates an online spectrum with small errors so that violations of the calculated envelopes take place.
40 Monitor Spectrum Envelope	This Analyzing Script monitors the online spectrum against the created spectrum envelopes.

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\Basic Spectrum Envelopes")

Name	Description
Visualize Spectrum Envelopes	This Monitoring View shows the original offline spectrum, the calculated spectrum envelopes, the current online spectrum and the three outputs of MonSpmEnv().

Offline Data
(found in "..\Example Offline Files\X-Tools Offline Data Files\Measurement Tasks\Basic Spectrum Envelopes")

Name	Description
Spectrum	The spectrum which has been generated via the Analyzing Model "10 Create Reference Spectrum".

7.4.7 Advanced Spectrum Envelopes

7.4.7.1 General

This chapter shows how to use the **Monitoring System** in order to create spectrum envelopes grapically. In addition, it shows how to use the created spectrum envelopes for monitoring of current online data against violations of these envelopes via the **Monitoring System** and/or the **Analyzing System** and it shows how the violations of spectrum envelopes can be visualized directly within the **Monitoring System**.

The Monitoring Views, Analyzing Models and Analyzing Scripts from this chapter can be tried out because the input data and the Monitoring Views, Analyzing Models and Analyzing Scripts are delivered together with ***X-Tools***. The input data is found within the "..\Server\User\Example Offline Files\X-Tools Offline Data Files\Measurement Tasks\Advanced Spectrum Envelopes" path of the ***X-Tools*** installation and must be loaded from there, whereas the used Analyzing Models, Analyzing Scripts and Monitoring Views are part of the example location which is delivered together with ***X-Tools***. The "Measurement Tasks\Advanced Spectrum Envelopes" subdirectory of each Configuration File type contains the Configuration Files to which the following descriptions refer (see also point 0).

7.4.7.2 Preparations

The following example uses a configurable wave whose frequency and order spectrums are calculated and monitored afterwards. Start the following Analyzing Script in order to define the properties of the wave:

- ..\Measurement Tasks\Advanced Spectrum Envelopes\05 Prepare Wave 1

This Analyzing Script adds 17 scalar data to the **MDS Explorer**. Start the following Analyzing Model in order to calculate an online data wave on base of the present scalar data:

- ..\Measurement Tasks\Advanced Spectrum Envelopes\Online\10 Generate Wave

This Analyzing Model calculates the wave plus the matching speed information. Based on the calculated wave, frequency and order spectrums can be calculated via the following Analyzing Models:

- ..\Measurement Tasks\Advanced Spectrum Envelopes\Online\20 Calculate Frequency Spectrum
- ..\Measurement Tasks\Advanced Spectrum Envelopes\Online\20 Calculate Order Spectrum

After these 3 Analyzing Models have been started, there are 6 online data present within the MDX.

There are also Analyzing Models available which calculate offline data. Start each of the following Analyzing Models (wait until the stop of each started Analyzing Model before starting of the next one):

- ..\Measurement Tasks\Advanced Spectrum Envelopes\Offline\10 Generate Wave
- ..\Measurement Tasks\Advanced Spectrum Envelopes\Offline\20 Calculate Frequency Spectrum
- ..\Measurement Tasks\Advanced Spectrum Envelopes\Offline\20 Calculate Order Spectrum

After these 3 Analyzing Models have been processed, there are 6 offline data present within the MDX.

7.4.7.3 Creation of Spectrum Envelopes

In order to create a (frequency or order) spectrum envelope, the input (frequency or order) spectrum must be available. The following screenshot shows the visualization of an example frequency spectrum. This example data is calculated by the three offline Analyzing Models from above:

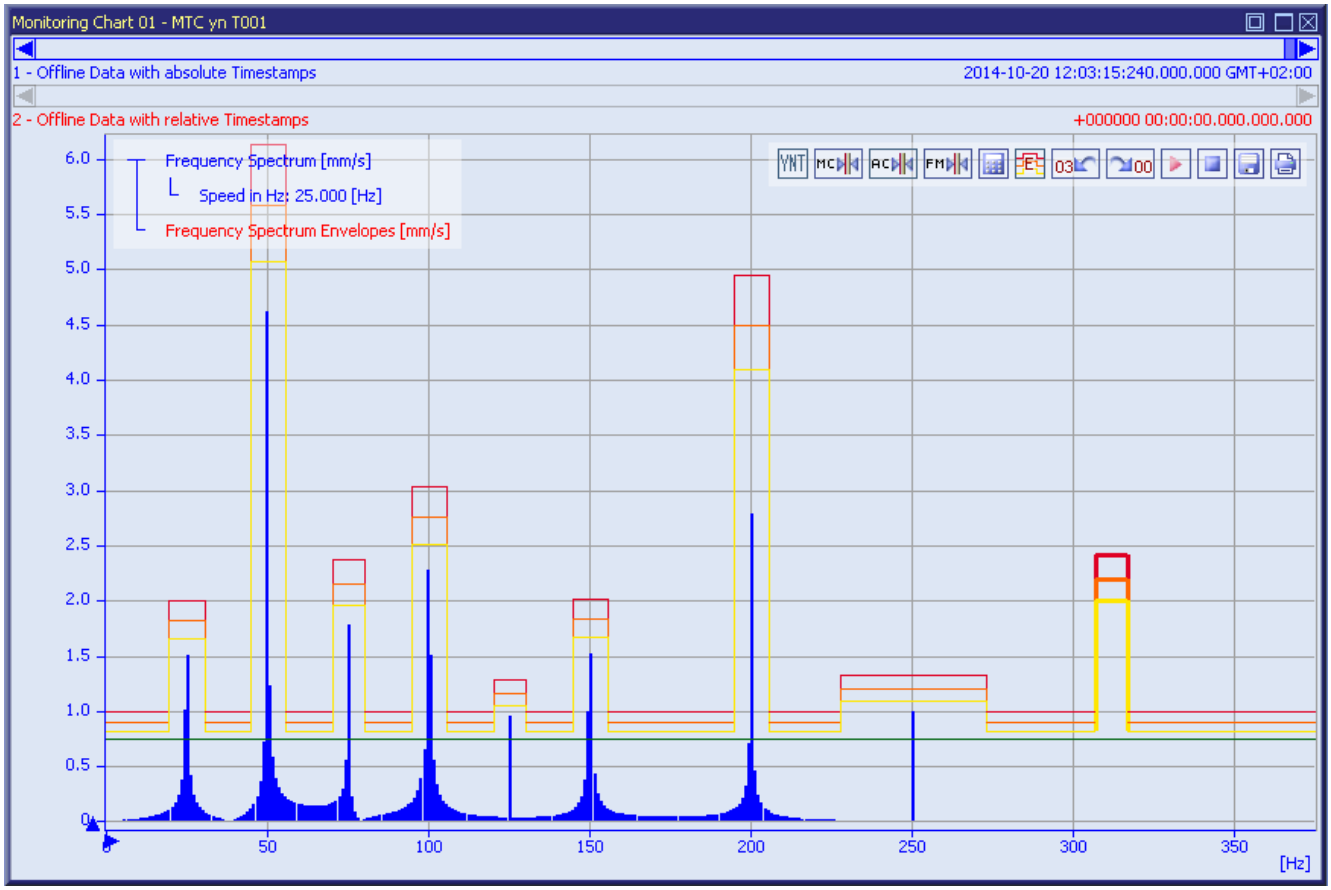


Figure 102: Example of the Input Data for an advanced Spectrum Envelope

The screenshot has been made after the “Edit Envelopes” mode was turned on via the Switch Envelope Mode toolbar button ( ). It can be seen that the “Edit Envelopes” mode is active because the E is shown by this toolbar button.

The (reference, warning, critical and defect) base bands can be moved up and down in order to add/remove alarm bands automatically. Alarm bands can be added and removed via the context menu and the present alarm bands can be changed via drag&drop.

The **Configure Envelopes** and **Configure Alarm Band** dialogs can be called via the context menu of the Switch Envelope Mode toolbar button and/or via a double-click onto the base band/alarm band. They can be used in order to define exact values for all properties of the base bands and of the alarm bands manually. In case of frequency spectrum envelopes, the **Configure Alarm Band** dialog also can be used in order to define whether an alarm band is at an absolute x-position or whether its x-position is relative to the current speed. In case relative alarm bands shall be used, the reference speed must be added to the legend entry of the reference frequency spectrum before the “Edit Envelopes” mode is entered.

In order to store the created spectrum envelope the context menu of the Switch Envelope Mode toolbar button must be called and Create Envelope Data must be chosen. This adds the created spectrum envelope to the **MDS Explorer** with the name which is specified within the **Configure Envelopes** dialog. In case the spectrum envelope shall be saved permanently, it must be stored to the disk via the **MDS Explorer**.

7.4.7.4 Monitoring of online Spectrum Envelopes

After the necessary envelopes have been created, they can be used for monitoring of the currently present spectrum data. For the following examples, load the spectrum envelopes which are delivered together with ***X-Tools*** from the following folder:

- ..\X-Tools Offline Data Files\Measurement Tasks\Advanced Spectrum Envelopes

The following screenshot shows the Analyzing Model “Online\30 Monitor Frequency Spectrum” from the example location which uses the above envelopes in order to monitor the current online spectrum:

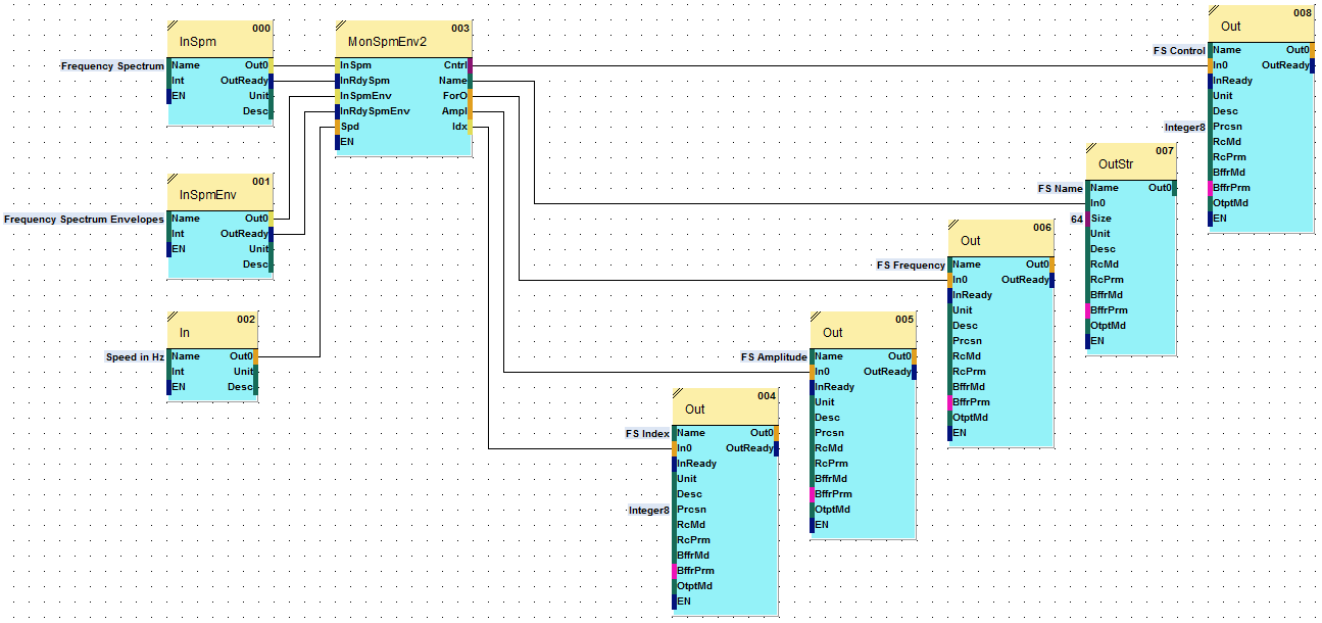


Figure 103: Example of an Analyzing Model for the monitoring of the current online Spectrum

The same task can be solved via an Analyzing Script also - the following box shows the Analyzing Script “Online\30 Monitor Frequency Spectrum” from the example location:

```
var main()
{
    var     Spm;
    var     SpmRdy;
    var     SpmEnv;
    var     SpmEnvRdy;
    var     Name;
    var     Frequency;
    var     Amplitude;
    var     Index;


    Spm    = InSpm( "Frequency Spectrum", "Default", true, SpmRdy );
    SpmEnv = InSpmEnv( "Frequency Spectrum Envelopes", "Default", true, SpmEnvRdy );
    ~'FS Control' = MonSpmEnv2( Spm, SpmRdy, SpmEnv, SpmEnvRdy, ~'Speed in Hz', true,
                                Name, Frequency, Amplitude, Index );

    ~'FS Name'      = Name;
    ~'FS Frequency' = Frequency;
    ~'FS Amplitude' = Amplitude;
    ~'FS Index'     = Index;

}
```

Figure 104: Example of an Analyzing Script for the monitoring of the current online Spectrum

Whenever a violation of the specified envelopes is detected, **MonSpmEnv2()** creates an according log entry.

Start either the Analyzing Model or the Analyzing Script in order to start to monitor the current online frequency spectrum against the loaded frequency spectrum envelope.

Monitoring of order spectrums works in exactly the same way, except that the speed information is not relevant for order spectrums at any moment. Start either the Analyzing Model or the Analyzing Script “Online\30 Monitor

Order Spectrum” in order to start to monitor the current online order spectrum against the loaded order spectrum envelope.

7.4.7.4.1 Example

Open the Monitoring View “Online\20 View online Violations”. This Monitoring View shows the online (frequency and order) spectrums together with their loaded spectrum envelopes. In addition, there are items at the right which allow to manipulate the online wave (which changes the calculated spectrums and their violations of the spectrum envelopes) and which show the current status from **MonSpmEnv2()**.

The following screenshot shows an example of this Monitoring View with the default configuration of the wave:

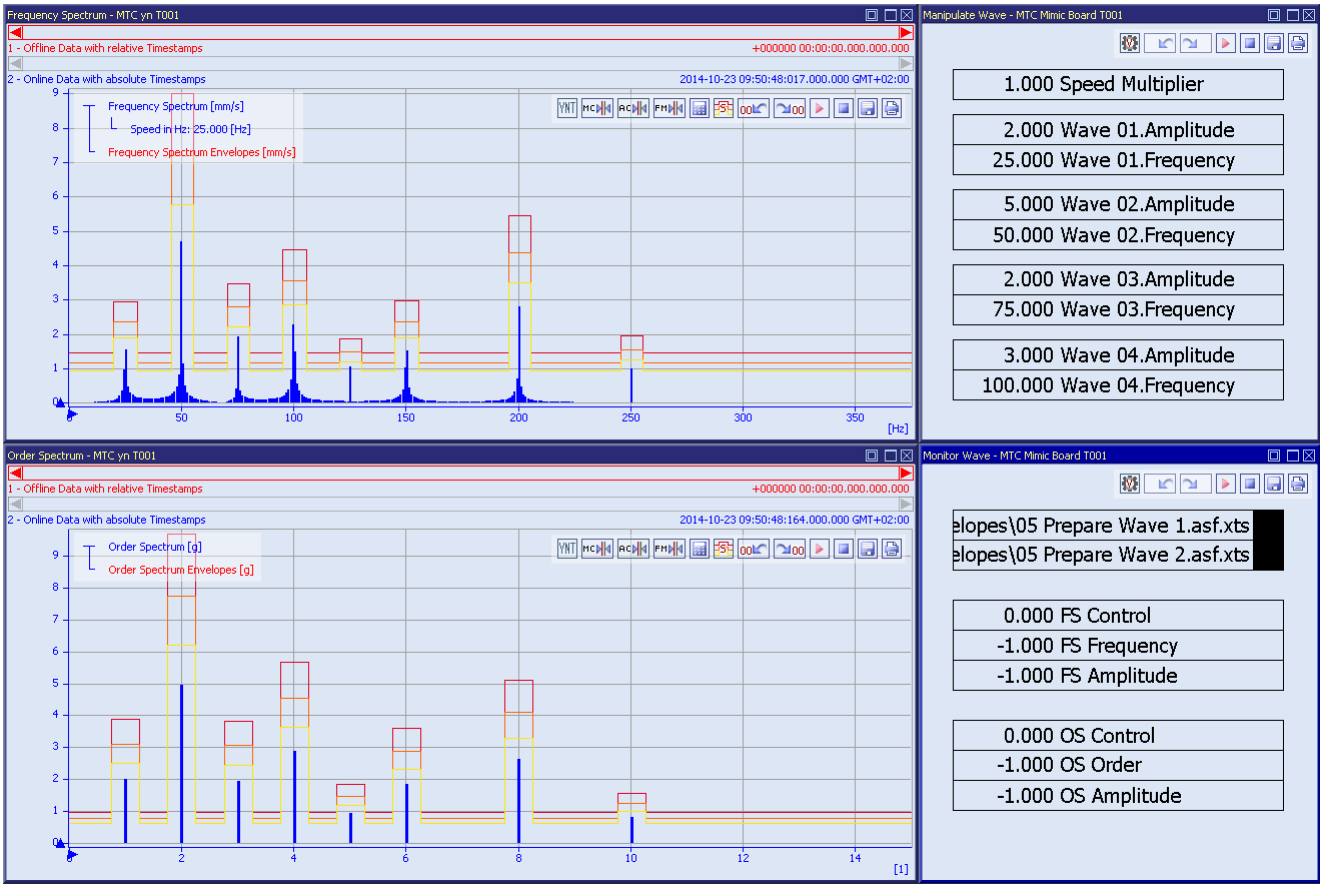


Figure 105: The Monitoring View “20 View online Violations” with the default Wave

Both Monitoring Charts at the left are in “Show Violations” mode, which can be seen from the “S” within the Switch Envelope Mode button. As none of the present peaks violate any of the spectrum envelopes, there are no violations shown at this moment.

Start the Analyzing Script “05 Prepare Wave 2” via the bottom right MTC Mimic Board T001. This Analyzing Script modifies several parameters of the wave, including the current speed which is used by the relative alarm bands of the frequency spectrum envelope. As a result, the position of all relative alarm bands within the frequency spectrum envelope is updated automatically.

The following screenshot shows an example after the modification of the wave:

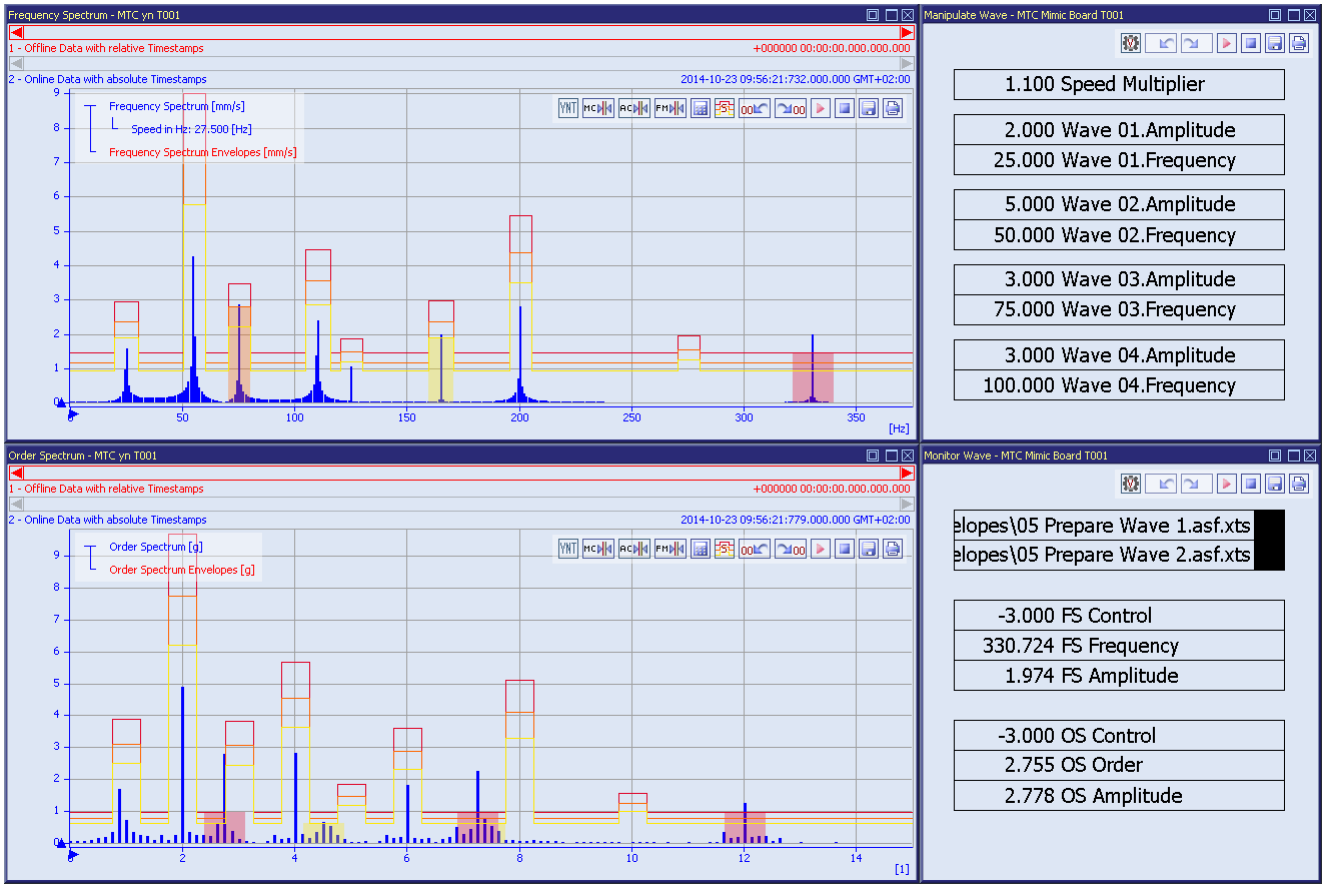


Figure 106: The Monitoring View "Online\20 View online Violations" with a modified Wave

At this screenshot, both Monitoring Charts at the left show several violations of their spectrum envelopes. In addition, the MTC Mimic Board T001 at the bottom right shows the current status from **MonSpmEnv2()**.

Change additional parameters of the wave via the top right MTC Mimic Board T001 in order to add, modify or remove violations of the present spectrum envelopes.

7.4.7.5 Monitoring of online Amplitudes

Start the Analyzing Script “05 Prepare Wave 1” in order to restore the default configuration of the used wave.

The following screenshot shows the Analyzing Model “Online\30 Monitor Amplitudes” from the example location which uses the above envelopes in order to monitor several amplitudes of the current online spectrum:

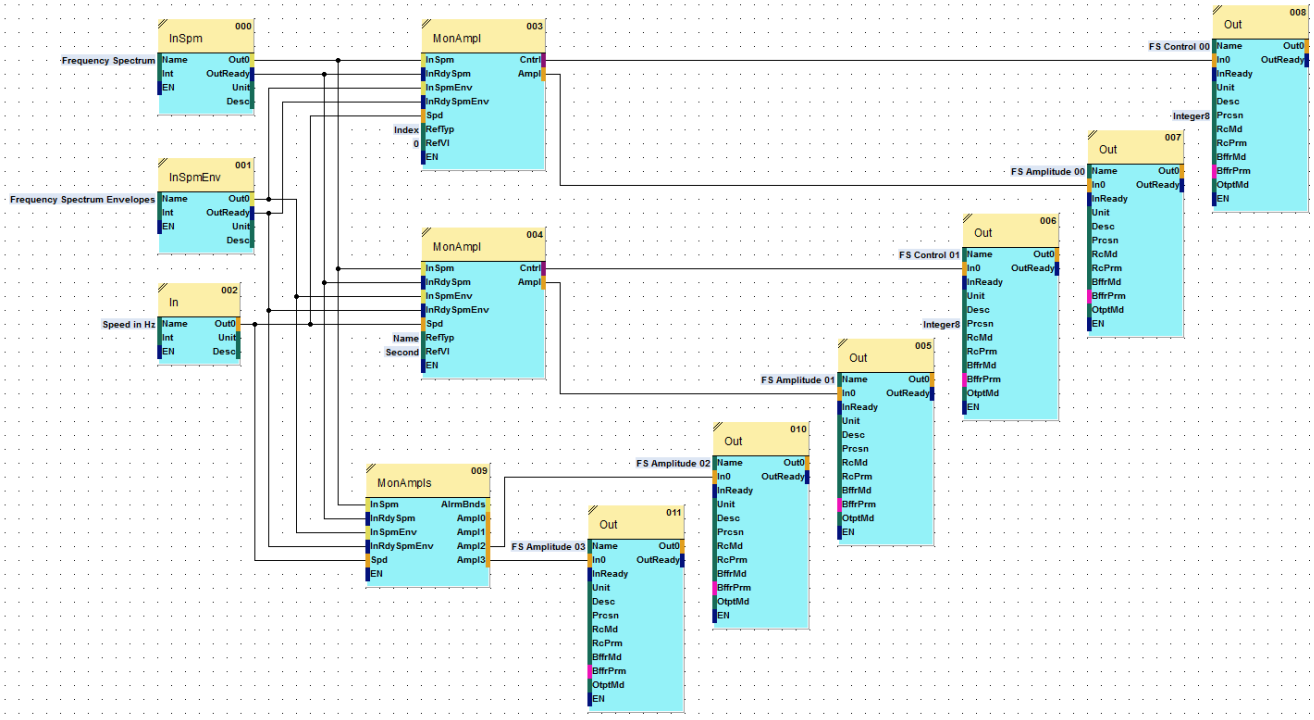


Figure 107: Example of an Analyzing Model for the monitoring of Amplitudes

The same task can be solved via an Analyzing Script also - the following box shows the Analyzing Script “Online\35 Monitor Amplitudes” from the example location:

```
var main()
{
    var     Spm;
    var     SpmRdy;
    var     SpmEnv;
    var     SpmEnvRdy;
    var     Amplitude00;
    var     Amplitude01;
    var     Amplitude02;
    var     Amplitude03;
    var     Dummy[2];


    Spm    = InSpm( "Frequency Spectrum", "Default", true, SpmRdy );
    SpmEnv = InSpmEnv( "Frequency Spectrum Envelopes", "Default", true, SpmEnvRdy );
    ~'FS Control 00' = MonAmpl( Spm, SpmRdy, SpmEnv, SpmEnvRdy, ~'Speed in Hz',
                                "Index", 0, true, Amplitude00 );
    ~'FS Control 01' = MonAmpl( Spm, SpmRdy, SpmEnv, SpmEnvRdy, ~'Speed in Hz',
                                "Name", "Second", true, Amplitude01 );
    ~'FS Amplitude 00' = Amplitude00;
    ~'FS Amplitude 01' = Amplitude01;

    MonAmpls( Spm, SpmRdy, SpmEnv, SpmEnvRdy, ~'Speed in Hz',
              true, Dummy[0], Dummy[1], Amplitude02, Amplitude03; );

    Out( "FS Amplitude 02", Amplitude02 );
    Out( "FS Amplitude 03", Amplitude03 );

}
```

Figure 108: Example of an Analyzing Script for the monitoring of Amplitudes

The Analyzing Functions **MonAmpl()** and **MonAmpls()** are being used for the monitoring of Amplitudes. **MonAmpl()** monitors a single alarm band and returns whether the current spectrum violates this alarm band as well as the currently present maximal amplitude below this alarm band. **MonAmpls()** returns only the maximal amplitudes which are present below each of the present alarm bands.

Start either the Analyzing Model or the Analyzing Script in order to start to monitor the amplitudes of the current online frequency spectrum.

Monitoring of the amplitudes of order spectrums works in exactly the same way, except that the speed information is not relevant for order spectrums at any moment.

7.4.7.5.1 Example

Open the Monitoring View “35 View online Amplitude Trends”. This Monitoring View shows an online spectrum together with its loaded spectrum envelope. In addition, there are items at the right which allow to manipulate the online wave (which changes the calculated spectrums and their violations of the spectrum envelopes) and which show the current status from **MonAmpl()** and **MonAmpls()**.

The following screenshot shows an example of this Monitoring View with the default configuration of the wave:

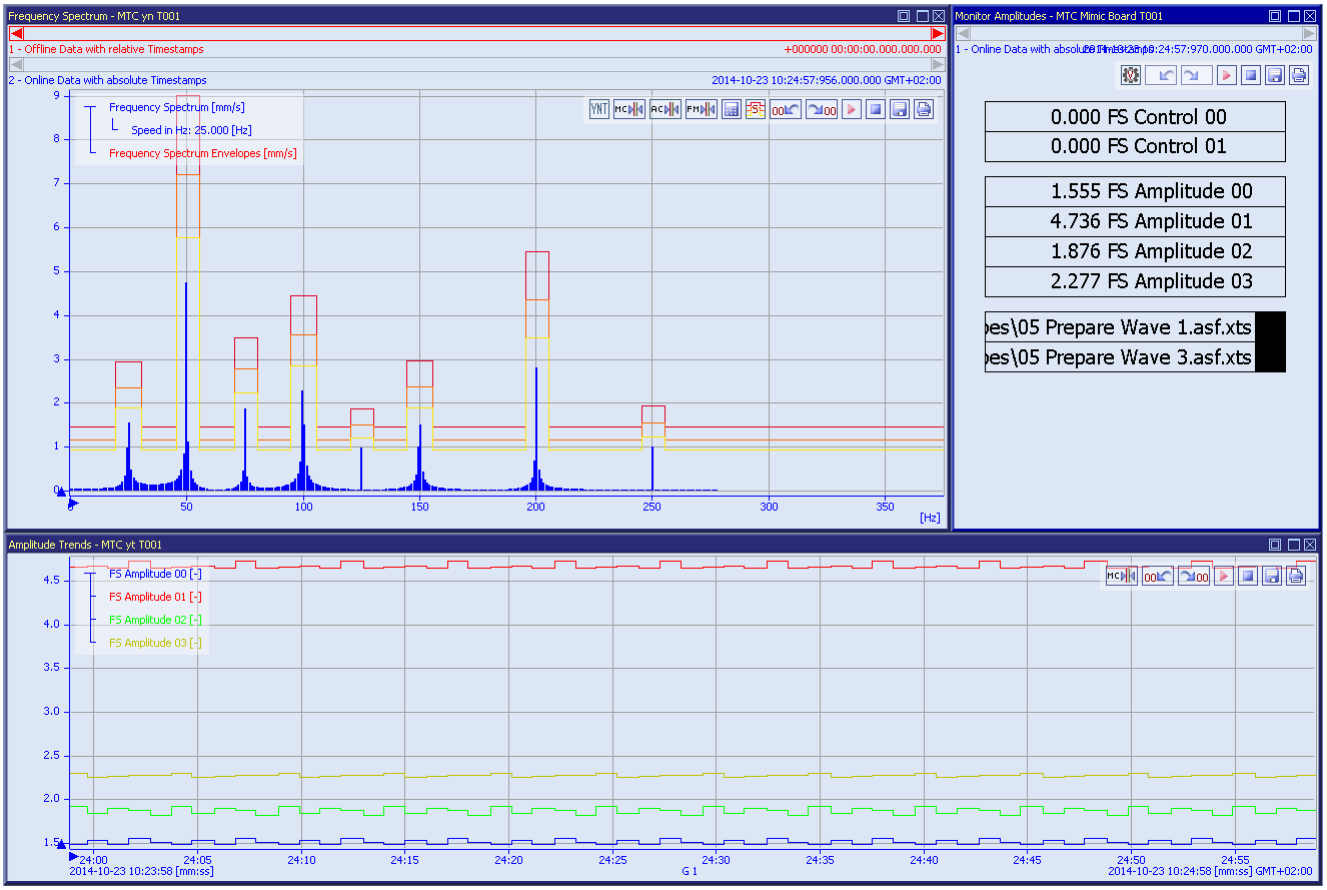

Frequency Spectrum - MTC yn T001
1 - Offline Data with relative Timestamps
+000000 00:00:00.000.000.000
2 - Online Data with absolute Timestamps
2014-10-23 10:24:57:956.000.000 GMT+02:00
Frequency Spectrum [mm/s]
Speed in Hz: 25.000 [Hz]
Frequency Spectrum Envelopes [mm/s]
[Hz]
Monitor Amplitudes - MTC Mimic Board T001
0.000 FS Control 00
0.000 FS Control 01
1.555 FS Amplitude 00
4.736 FS Amplitude 01
1.876 FS Amplitude 02
2.277 FS Amplitude 03
Amplitude Trends - MTC yt T001
FS Amplitude 00 [-]
FS Amplitude 01 [-]
FS Amplitude 02 [-]
FS Amplitude 03 [-]
2014-10-23 10:23:58 [mm:ss]
2014-10-23 10:24:58 [mm:ss] GMT+02:00

Figure 109: The Monitoring View “35 View online Amplitude Trends” with the default Wave

The Monitoring Chart at the top left is in “Show Violations” mode, which can be seen from the “S” within the Switch Envelope Mode button. As none of the present peaks violate the spectrum envelope, there are no violations shown at this moment.

Start the Analyzing Script “05 Prepare Wave 3” via the top right MTC Mimic Board T001. This Analyzing Script modifies several parameters of the wave, including the current speed which is used by the relative alarm bands of the frequency spectrum envelope.

The following screenshot shows an example after the modification of the wave:

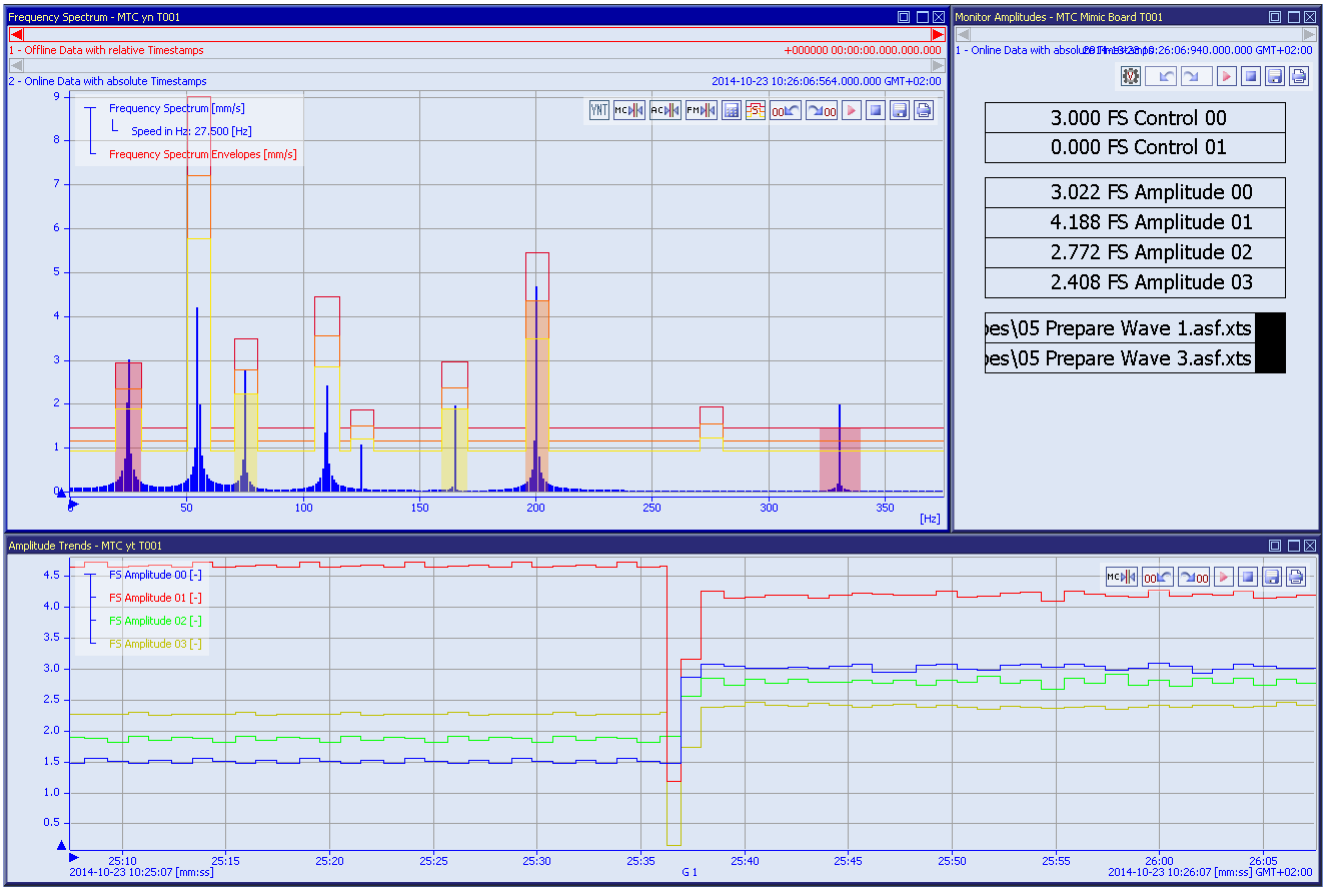

Frequency Spectrum - MTC yn T001
1 - Offline Data with relative Timestamps
+000000 00:00:00.000.000.000
2 - Online Data with absolute Timestamps
2014-10-23 10:26:06:564.000.000 GMT+02:00
Frequency Spectrum [mm/s]
Speed in Hz: 27.500 [Hz]
Frequency Spectrum Envelopes [mm/s]
[Hz]
Monitor Amplitudes - MTC Mimic Board T001
3.000 FS Control 00
0.000 FS Control 01
3.022 FS Amplitude 00
4.188 FS Amplitude 01
2.772 FS Amplitude 02
2.408 FS Amplitude 03
pes\05 Prepare Wave 1.asf.xts
pes\05 Prepare Wave 3.asf.xts
Amplitude Trends - MTC yt T001
FS Amplitude 00 [-]
FS Amplitude 01 [-]
FS Amplitude 02 [-]
FS Amplitude 03 [-]
2014-10-23 10:25:07 [mm:ss]
G 1
2014-10-23 10:26:07 [mm:ss] GMT+02:00

Figure 110: The Monitoring View “35 View online Amplitude Trends” with a modified Wave

At this screenshot, the Monitoring Chart at the top left shows several violations of its spectrum envelope. In addition, the MTC Mimic Board T001 at the top right shows the current status from **MonAmpl()** and **MonAmpls()**.

The bottom Monitoring Chart shows the amplitude trend of four alarm bands. About in the middle of the displayed time interval the Analyzing Script “05 Prepare Wave 3” has changed the parameters of the wave.

Change additional parameters of the wave via the top right MTC Mimic Board T001 in order to add, modify or remove violations of the present spectrum envelope.

7.4.7.6 Monitoring of offline Data

All of the Analyzing Functions from the previous chapters also can be used in order to monitor offline spectrum envelopes. According examples of Analyzing Models, Analyzing Scripts and Monitoring Views are found within the example location (see also point 7.4.7.7).

The following screenshot shows an example of the Monitoring View “20 View offline Violations”:

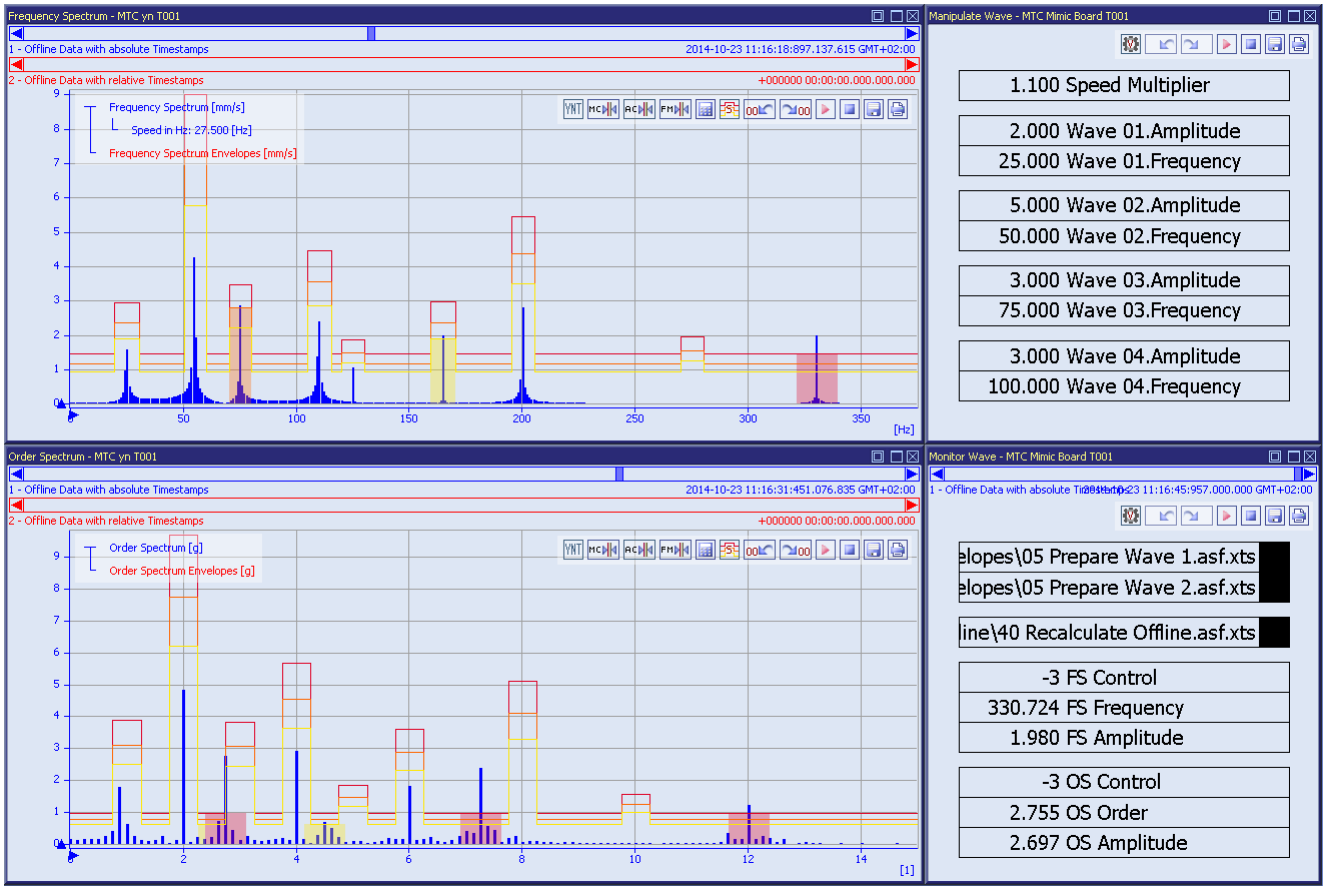


Figure 111: The Monitoring View “20 View offline Violations”

The parameters within the upper right Monitoring Chart can be modified in order to change the used wave. Afterwards, the Analyzing Script “40 Recalculate Offline” can be called via the bottom right Monitoring Chart in order to recalculate the offline wave, the offline (frequency and order) spectrums and the violations of the spectrum envelopes.

The following screenshot shows an example of the Monitoring View “35 View offline Amplitude Trends”:

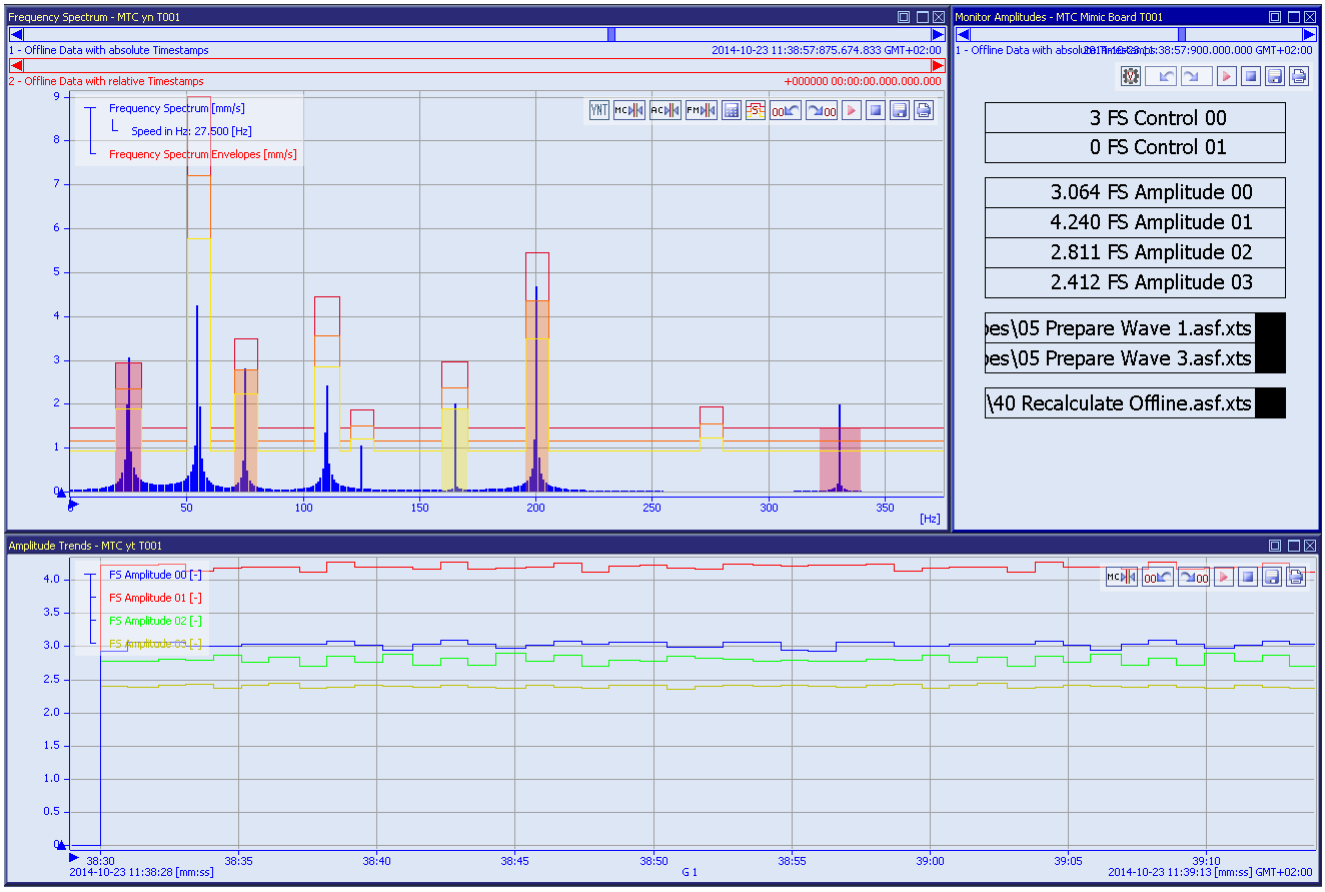

Frequency Spectrum - MTC yn T001
1 - Offline Data with absolute Timestamps
2014-10-23 11:38:57:875.674.833 GMT+02:00
2 - Offline Data with relative Timestamps
+000000 00:00:00.000.000.000
Frequency Spectrum [mm/s]
Speed in Hz: 27.500 [Hz]
Frequency Spectrum Envelopes [mm/s]
[Hz]
Monitor Amplitudes - MTC Mimic Board T001
3 FS Control 00
0 FS Control 01
3.064 FS Amplitude 00
4.240 FS Amplitude 01
2.811 FS Amplitude 02
2.412 FS Amplitude 03
\40 Recalculate Offline.asf.xts
Amplitude Trends - MTC yt T001
FS Amplitude 00 [-]
FS Amplitude 01 [-]
FS Amplitude 02 [-]
2014-10-23 11:38:28 [mm:ss]
G 1
2014-10-23 11:39:13 [mm:ss] GMT+02:00

Figure 112: The Monitoring View “35 View offline Amplitude Trends”

The parameters within the upper right Monitoring Chart can be modified in order to change the used wave. Afterwards, the Analyzing Script “40 Recalculate Offline” can be called via the upper right Monitoring Chart in order to recalculate the offline wave, the offline (frequency and order) spectrums and the amplitude trends.

7.4.7.7 Summary

The following Configuration Files are being used by this chapter:

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Advanced Spectrum Envelopes")

Name	Description
Offline\10 Generate Wave	This Analyzing Model uses the current values of the prepared scalar data in order to calculate an offline data wave and the according speed information.
Offline\20 Calculate Frequency Spectrum	This Analyzing Model calculates the frequency spectrum for the offline data wave.
Offline\20 Calculate Order Spectrum	This Analyzing Model calculates the order spectrum for the offline data wave.
Offline\30 Monitor Frequency Spectrum	This Analyzing Model monitors the offline frequency spectrum against the created frequency spectrum envelopes.
Offline\30 Monitor Order Spectrum	This Analyzing Model monitors the offline order spectrum against the created order spectrum envelopes.
Offline\35 Monitor Amplitudes	This Analyzing Model creates the amplitude trends for several alarm bands of the offline frequency spectrum envelope.
Online\10 Generate Wave	This Analyzing Model uses the current values of the prepared scalar data in order to calculate an online data wave and the according speed information.
Online\20 Calculate Frequency Spectrum	This Analyzing Model calculates the frequency spectrum for the online data wave.
Online\20 Calculate Order Spectrum	This Analyzing Model calculates the order spectrum for the online data wave.
Online\30 Monitor Frequency Spectrum	This Analyzing Model monitors the online frequency spectrum against the created frequency spectrum envelopes.
Online\30 Monitor Order Spectrum	This Analyzing Model monitors the online order spectrum against the created order spectrum envelopes.
Online\35 Monitor Amplitudes	This Analyzing Model creates the amplitude trends for several alarm bands of the online frequency spectrum envelope.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Advanced Spectrum Envelopes")

Name	Description
05 Prepare Wave 1	This Analyzing Script prepares several scalar data so that they can be used in order to produce a certain wave.
05 Prepare Wave 2	This Analyzing Script prepares several scalar data so that they can be used in order to produce a certain wave.
05 Prepare Wave 3	This Analyzing Script prepares several scalar data so that they can be used in order to produce a certain wave.
Offline\30 Monitor Frequency Spectrum	This Analyzing Script monitors the offline frequency spectrum against the created frequency spectrum envelopes.
Offline\30 Monitor Order Spectrum	This Analyzing Script monitors the offline order spectrum against the created order spectrum envelopes.
Offline\35 Monitor Amplitudes	This Analyzing Script creates the amplitude trends for several alarm bands of the offline frequency spectrum envelope.
Offline\40 Recalculate Offline	This Analyzing Script recalculates all of the offline data which is being used by this measurement task.
Online\30 Monitor Frequency Spectrum	This Analyzing Script monitors the online frequency spectrum against the created frequency spectrum envelopes.
Online\30 Monitor Order Spectrum	This Analyzing Script monitors the online order spectrum against the created order spectrum envelopes.
Online\35 Monitor Amplitudes	This Analyzing Script creates the amplitude trends for several alarm bands of the online frequency spectrum envelope.

Monitoring Views

(found in "..\Monitoring View Files\Measurement Tasks\Advanced Spectrum Envelopes")

Name	Description
10 Edit Frequency Spectrum Envelope	This Monitoring View can be used in order to edit the frequency spectrum envelope which is being used by this measurement task.
10 Edit Order Spectrum Envelope	This Monitoring View can be used in order to edit the order spectrum envelope which is being used by this measurement task.
20 View offline Violations	This Monitoring View can be used in order to show the violations of offline (frequency and order) spectrum envelopes.
20 View online Violations	This Monitoring View can be used in order to show the violations of online (frequency and order) spectrum envelopes.
35 View offline Amplitude Trends	This Monitoring View can be used in order to show the amplitude trends of an offline frequency spectrum.
35 View online Amplitude Trends	This Monitoring View can be used in order to show the amplitude trends of an online frequency spectrum.

Offline Data

(found in "..\Example Offline Files\X-Tools Offline Data Files\Measurement Tasks\Advanced Spectrum Envelopes")

Name	Description
Frequency Spectrum Envelopes	The frequency spectrum envelopes which match the configuration of the used Analyzing Models and Analyzing Scripts.
Order Spectrum Envelopes	The order spectrum envelopes which match the configuration of the used Analyzing Models and Analyzing Scripts.

7.4.8 Automatic loading of Data

7.4.8.1 General

This chapter shows how to use the **Analyzing System** and the Analyzing Function **LoadData()** in order to load offline data automatically from offline data files. The different use cases for the loading of offline data files are handled by the following chapters.

The Analyzing Scripts and Loading Profiles from this chapter can be tried out because the to-be-loaded data, the Analyzing Scripts and the Loading Profiles are delivered together with ***X-Tools***. The to-be-loaded data is found within the "..\Server\User\Example Offline Files\X-Tools Offline Data Files" path of the ***X-Tools*** installation, whereas the used Analyzing Scripts and Loading Profiles are part of the example location which is delivered together with ***X-Tools***. The "Measurement Tasks\Loading of Data" subdirectory of each Configuration File type contains the Configuration Files to which the following descriptions refer (see also point 7.4.8.5).

7.4.8.2 Basic loading of offline Data

The most simple usage of **LoadData()** is to load all files from a certain directory and all of its sub-directories. The Analyzing Script "Basic Loading" from the example location shows an example for this use case:

```
var main()
{
    // the location "X-Tools Offline Data Files" must point to the following directory
    // of the X-Tools installation:
    // ..\Server\User\Example Offline Files\X-Tools Offline Data Files

    // the following command loads all offline data from the specified directory
    // and all of its sub-directories
    LoadData("X-Tools Offline Data Files://ION AnalogInput T001");
}
```

7.4.8.3 Loading of offline Data Files from a certain Period of Time

Another use case for the loading of offline data files is to automatically load a certain period of time of some data. The Analyzing Script "Load Period of Time" from the example location shows an example for this use case:

```
var main()
{
    // the location "X-Tools Offline Data Files" must point to the following directory
    // of the X-Tools installation:
    // ..\Server\User\Example Offline Files\X-Tools Offline Data Files

    var    StartTime,EndTime;

    StartTime = 1244110988;    // 2009-06-04 12:23:09
    EndTime   = 1244110994;    // 2009-06-04 12:23:15

    // we load 6 seconds from the middle of this data
    LoadData("X-Tools Offline Data Files://ION AnalogInput T001",
             "ION AnalogInput T001 CH01",
             "Append","","Each Value","",
             "Minimum Time and Maximum Time",StartTime,0,EndTime);
}
```

7.4.8.4 Loading with Loading Profiles and Loading Modes

Loading Profiles and all of their contained filter functionality can be used in order to apply filters during the loading of data. The Analyzing Script “Load with Loading Profile, Loading Modes” from the example location shows the result of loading with a Loading Profile which has loading modes configured:

```
var main()
{
    // the location "X-Tools Offline Data Files" must point to the following directory
    // of the X-Tools installation:
    // ..\Server\User\Example Offline Files\X-Tools Offline Data Files

    // the following command loads data from the specified directory and applies
    // the loading modes which are configured within the specified Loading Profile
    LoadData("X-Tools Offline Data Files://ION AnalogInput T001",
             "","Append",
             "://Measurement Tasks\\Loading of Data\\Loading Modes.lpf.xts");
}
```

7.4.8.5 Loading with Loading Profiles and Loading Times

Loading Profiles and all of their contained filter functionality can be used in order to apply filters during the loading of data. The Analyzing Script “Load with Loading Profile, Loading Times” from the example location shows the result of loading with a Loading Profile which has loading times configured:

```
var main()
{
    // the location "X-Tools Offline Data Files" must point to the following directory
    // of the X-Tools installation:
    // ..\Server\User\Example Offline Files\X-Tools Offline Data Files

    // the following commands load data from the specified directory and applies
    // the loading times which are configured within the specified Loading Profile
    LoadData("X-Tools Offline Data Files://ION AnalogInput T001",
             "","Append",
             "://Measurement Tasks\\Loading of Data\\Loading Times.lpf.xts");
}
```

7.4.8.6 Loading of the last Minutes of the Data from a Storage Profile

The Analyzing Script “Load with Storage Profile, Period of Time” from the example location shows an example of how the last few minutes of the data from a certain Storage Profile can be loaded:

```
var main()
{
    // the location "Offline Files" also must be used by the following Storage Profile:
    // - Measurement Tasks\Loading of Data\IFN AI

    // the following Configuration Files must be running:
    // - Interface Profile Files\Measurement Tasks\Loading of Data\Simulation T001
    // - Device Profile Files\Measurement Tasks\Loading of Data\IFN AI
    // - Storage Profile Files\Measurement Tasks\Loading of Data\IFN AI

    var StartTime;
    var EndTime;

    // the current data may be written still, so let's load until "a little bit before now"
    EndTime = GetTime() - 100;

    // we want to load a total of 5 minutes
    StartTime = EndTime - 300;

    // load the data from the specified time interval
    LoadData("://Measurement Tasks\\Loading of Data\\IFN AI.spf.xts",
             "","Append","","Each Value","",
             "Minimum Time and Maximum Time",StartTime,0,EndTime);
}
```

7.4.8.7 Loading of the last Files from a Storage Profile (once)

The Analyzing Script “Load with Storage Profile, Latest Files, once” from the example location shows an example of how the last stored files from a certain Storage Profile can be loaded:

```
var main()
{
    // the location "Offline Files" also must be used by the following Storage Profile:
    // - Measurement Tasks\Loading of Data\IFN AI

    // the following Configuration Files must be running:
    // - Interface Profile Files\Measurement Tasks\Loading of Data\Simulation T001
    // - Device Profile Files\Measurement Tasks\Loading of Data\IFN AI
    // - Storage Profile Files\Measurement Tasks\Loading of Data\IFN AI

    // load the data from the specified time interval
    LoadData("://Measurement Tasks\\Loading of Data\\IFN AI.spf.xts",
             "","Overwrite","","Each Value","",
             "Load Latest Files");
}
```

7.4.8.8 Loading of the last Files from a Storage Profile (automatically)

The Analyzing Script “Load with Storage Profile, Latest Files, automatic” from the example location shows an example of how to load the last stored files from a certain Storage Profile automatically, all the time while the Storage Profile is running (and producing additional offline data):

```
var main()
{
    // the location "Offline Files" also must be used by the following Storage Profile:
    // - Measurement Tasks\Loading of Data\IFN AI

    // the following Configuration Files must be running:
    // - Interface Profile Files\Measurement Tasks\Loading of Data\Simulation T001
    // - Device Profile Files\Measurement Tasks\Loading of Data\IFN AI
    // - Storage Profile Files\Measurement Tasks\Loading of Data\IFN AI

    while(1)
    {
        // with the following settings, WaitForEDT() comes back each 10 seconds,
        // thus LoadData() also is being called once each 10 seconds
        WaitForEDT("2000-01-01","00:00:00",10);

        // load the data from the specified time interval
        LoadData("://Measurement Tasks\\Loading of Data\\IFN AI.spf.xts",
                 "","Overwrite","","Each Value","",
                 "Load Latest Files");
    }
}
```

The Monitoring View “Automatic Update of the t-Axes” from the example location can be used in order to see how the loaded offline data is taken over automatically by a Monitoring View. In order to enable the automatic update of the Monitoring View, the option “Rescale t-Axes after Action” must be configured to “Yes” (either within the **Chart Options** dialog or via the context menu of each t-axis).

7.4.8.9 Summary

The following Configuration Files are being used by this chapter:

Interface Profiles
(found in “..\Interface Profile Files\Measurement Tasks\Loading of Data”)

Name	Description
Simulation T001	This Interface Profile enables the simulated devices which are being used for the generation of the to-be-stored data.

Device Profiles
(found in “..\Device Profile Files\Measurement Tasks\Loading of Data”)

Name	Description
IFN AI	This Device Profile provides the online data from a simulated IFN AI.

Monitoring Views
(found in “..\Monitoring View Files\Measurement Tasks\Loading of Data”)

Name	Description
Automatic Update of the t-Axes	This Monitoring View is configured to automatically update its **Monitoring Charts** whenever **LoadData()** has loaded the latest files of the currently running Storage Profile again.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Loading of Data")

Name	Description
Basic Loading	This Analyzing Script shows a very easy method for the loading of offline data from a certain directory.
Load Period of Time	This Analyzing Script shows how to load a certain amount of time of the offline data from a certain directory.
Load with Loading Profile, Loading Modes	This Analyzing Scripts shows how to use Loading Profiles in order to load data with the available loading modes.
Load with Loading Profile, Loading Times	This Analyzing Script shows how to use Loading Profiles in order to load data with configuring of the to-be-loaded time intervals.
Load with Storage Profile, Latest Files, automatic	This Analyzing Script shows how to use **LoadData()** in order to automatically load the latest offline data files of the specified Storage Profile.
Load with Storage Profile, Latest Files, once	This Analyzing Script shows how to use **LoadData()** in order to load the latest offline data files of the specified Storage Profile once.
Load with Storage Profile, Period of Time	This Analyzing Script shows how to load a certain amount of time of the offline data from a certain Storage Profile.

Storage Profiles
(found in "..\Storage Profile Files\Measurement Tasks\Loading of Data")

Name	Description
IFN AI	This Storage Profile stores the oline data from the simulated IFN AI.

Loading Profiles
(found in "..\Loading Profile Files\Measurement Tasks\Loading of Data")

Name	Description
Loading Modes	This Loading Profile configures the available loading modes for the loading of offline data.
Loading Times	This Loading Profile configures different time intervals for the loading of offline data.

7.4.9 Advanced parallel Storing

7.4.9.1 General

This chapter shows how to configure a Storage Profile for applications where it is necessary to store the same data in parallel multiple times, where each storage shall maintain its own, data-based start and stop triggers. An example use case could be:

- There is a plant (e.g. a steel mill) with different sections.
- At any time, multiple parts (e.g. coils) could be processed by the different sections of the plant.
- The Storage System shall produce one directory with offline data for each processed part. This directory is started at the moment when the part enters the first section (an according start trigger is caused at this moment) and stopped at the moment when the part leaves the last section (an according stop trigger is caused at this moment).
- When the start trigger occurs, an ID which uniquely identifies the currently processed part (the so-called "Storage ID") shall be added as part of the directory name.
- At the moment when the plant is started, there are no parts within the different sections. Afterwards, multiple parts may enter the plant (= multiple start events occur) before the first part leaves the last section again (= the first stop event occurs).
- In such a configuration, there is no direct relation between the start and stop trigger any more and the Storage Profile must be configured accordingly in order to be able to produce the desired offline data files.

7.4.9.2 Plant Simulation

The Analyzing Script “Plant” is used in order to simulate a plant with all of the described behavior:

- A start event occurs each 30 seconds. Thus, the simulation acts as if another part would enter the plant each 30 seconds.
- The plant is separated into 5 sections. Each section processes the part for about 15 seconds. After the processing, the part moves for about 5 seconds until it arrives at the next section. Thus, the total processing time of each part is about 100 seconds.
  - The simulation is decreasing a counter with a speed which is different in each section. In a real plant like a rolling mill, a similar measurement could be received from a temperature or thickness sensor.
  - Within the simulated plant, there are always 3-4 parts processed in parallel. As a result, the Storage Profile is also storing the current data to 3-4 offline data directories in parallel.
- The simulation uses the current time, in seconds since 2012-01-01, as unique storage ID. This value changes with each start event and is added to the name of each created directory.
- A stop event occurs each 30 seconds. After the start of the Analyzing Script, the first stop event occurs 100 seconds after the first start event - this is due to the fact that the first processed part must go through the whole plant first before it can cause a stop event.

The Monitoring View “Plant (Online)” can be used in order to visualize all of the data from the simulated plant. The following screenshot shows an example from this Monitoring View:

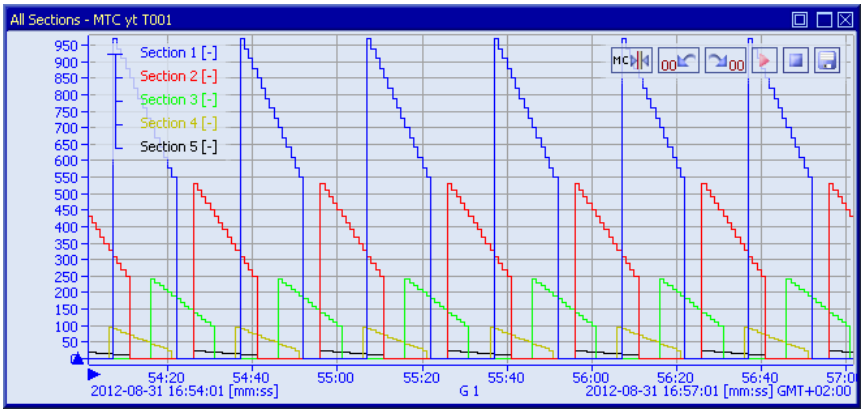


Figure 113: Example of the Plant Simulation for the advanced parallel storing

7.4.9.3 Storage Profile

The Storage Profile “Plant” is used in order to store all data from the simulated plant:

- A new offline data directory is created whenever the start event occurs.
- About 100 seconds of data are written to each offline data directory before the storing is ended through the (matching) stop trigger.

The Storage Profile can be started and stopped at any moment in time. The most important settings for this behavior of the Storage Profile are:

- “Storage Profile Parameters” table
  - Parallel Storage = Start a parallel Storage whenever another Start Event occurs
- “Stop Event” table
  - Minimal Start to Stop Interval = 95 seconds
    - the chosen interval should be short below the expected amount of time between the start and stop triggers which belong to one processed part

7.4.9.4 Summary

The following Configuration Files are being used by this chapter:

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\Advanced parallel Storing")

Name	Description
Plant (Online)	This Monitoring View displays all data from the simulated plant.
Plant (Offline)	This Monitoring View can display the offline data which has been stored by the Storage Profile "Plant". In case there is always one offline data directory loaded, the visualized data curves are identical because each offline data category contains the data from one processed part (and the simulated data is identical for each simulated part).

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Advanced parallel Storing")

Name	Description
Plant	This Analyzing Script simulates the basic komponents of the plant.

Storage Profiles
(found in "..\Storage Profile Files\Measurement Tasks\Advanced parallel Storing")

Name	Description
Plant	This Storage Profile uses the parallel storage mode "Start another parallel Storage whenever another Start Event occurs" in order create one offline data directory per processed part.

7.4.10 Charge based Analysis

7.4.10.1 General

This chapter shows how to configure an automatic analysis for charge-based processes. An example use case could be:

- There is a process (e.g. cooking of steel) which starts, operates on one charge, stops and starts with the next charge afterwards.
- There is a data which indicates the start of a charge (e.g. at the moment when the ore is put into the furnace).
- There is a data which indicates the stop of a charge (e.g. at the moment when the hot steel is tilt out of the furnace).
- After all data of one charge has been stored, an automatic analysis shall start.
- After the analysis of the current charge has been finished, the storing of the next charge is prepared and the analysis starts from the beginning again.

7.4.10.2 Plant Simulation

The Analyzing Script "10 Create Charges" is used in order to simulate a process with all of the described behavior:

- A start event occurs about each 30 - 45 seconds after the last stop trigger.
- A stop event occurs about each 10 - 15 seconds after the last start trigger.
- While a charge is being processed (= from the moment of the start trigger until the moment of the stop trigger), the simulated temperature is rising slightly from 50 °C upwards.
- After the stop of each charge, the simulated temperature drops to 30 °C.

The Monitoring View “Charges” can be used in order to visualize all of the data from the simulated process. The following screenshot shows an example from this Monitoring View:

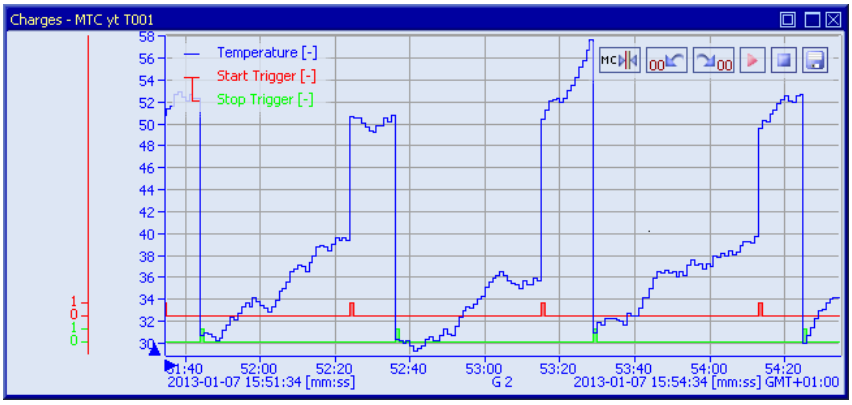


Figure 114: Example of the Process Simulation for the Charge based Analysis

7.4.10.3 Charge based Analysis

The example analysis is found within the Analyzing Script “20 Run Charge Processing” and performs the following tasks:

- All of the data from one charge is stored via the according Storage Profile “Store current Charge”.
- After the charge is ready, the Storage Profile stops and its created offline data are loaded automatically.
- The Analyzing Model “30 Analyse current Charge” is used in order to determine the maximal temperature which was present during the time interval of the charge.
- After the maximal temperature has been detemined by the first Analyzing Model, the Analyzing Model “40 Create Trends” is started in order to create a trend of the maximal temperatures of the last 10 charges.
- The Monitoring View “Overview” is updated automatically whenever the analysis of another charge has completed.

The following box shows the Analyzing Script “20 Run Charge Processing” from the example location:

```
var main()
{
    // start the Analyzing Script which simulates our charges
    Start( "://Measurement Tasks/Charge based Analysis/10 Create Charges.asf.xts", "DoNotWait" );

    // start the Analyzing Model which outputs the max temperature from the last charge as online data
    Start( "://Measurement Tasks/Charge based Analysis/10 Offline to Online.amf.xts", "DoNotWait" );

    while( 1 )
    {
        // store the data from the current charge
        Start( "://Measurement Tasks/Charge based Analysis/Store current Charge.spf.xts", "Wait" );

        // load the data from the last splitting
        LoadData( "://Measurement Tasks/Charge based Analysis/Store current Charge.spf.xts",
                  "","Overwrite","","Each Value","",
                  "Load Latest Files");

        // analyse the data from the current charge
        Start( "://Measurement Tasks/Charge based Analysis/30 Analyse current Charge.amf.xts", "Wait"
);

        // create the trends
        Start( "://Measurement Tasks/Charge based Analysis/40 Create Trends.amf.xts", "Wait" );
    }
}
```

Figure 115: Example of an Analyzing Script for the Charge-based processing

7.4.10.4 Summary

The following Configuration Files are being used by this chapter:

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\Charge based Analysis")

Name	Description
Charges	This Monitoring View displays all data from the simulated process.
Overview	This Monitoring View displays the data from the simulated process together with the results of the analysis.

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Charge based Analysis")

Name	Description
10 Offline to Online	This Analyzing Model outputs the last determined maximal temperature as online data.
30 Analyse current Charge	This Analyzing Model searches for the maximal value of the temperature data during the whole period of the charge.
40 Create Trends	This Analyzing Model creates a trend of the maximal temperature values of the last 10 charges.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Charge based Analysis")

Name	Description
10 Create Charges	This Analyzing Script simulates the charge-based process.

20 Run Charge Processing	This Analyzing Script performs the actual, charge-based analysis:
	• the simulation of the charge-based process is started
	• afterwards, the following tasks are running in an endless loop:
	  ○ a Storage Profile is started which stores exactly one charge and stops afterwards
	  ○ LoadData() is used in order to load the offline data which was created by the last run of the Storage Profile
	  ○ the Analyzing Model which analysis the current charge is started
	  ○ the Analyzing Model which creates the long-term trend is started

Storage Profiles
(found in "..\Storage Profile Files\Measurement Tasks\Charge based Analysis")

Name	Description
Store current Charge	This Storage Profile stores all of the data from one charge (from the moment of the start event until the moment of the stop event).

7.4.11 File Splitting based Analysis

7.4.11.1 General

This chapter shows how to configure an automatic analysis for file splitting based processes. An example use case could be:

- There is a process which runs constantly and which stores offline data constantly.
- The file splitting of Storage Profiles is used in order to limit the size of the created offline data files at the disk.
- Whenever another file splitting has completed, an automatic analysis shall start.

7.4.11.2 Input Data

The Analyzing Script “10 Create Source Data” is used in order to create some data which has some jitter and is moving updwards slowly.

The Monitoring View “Overview” can be used in order to visualize the simulated data. The following screenshot shows an example from this Monitoring View:

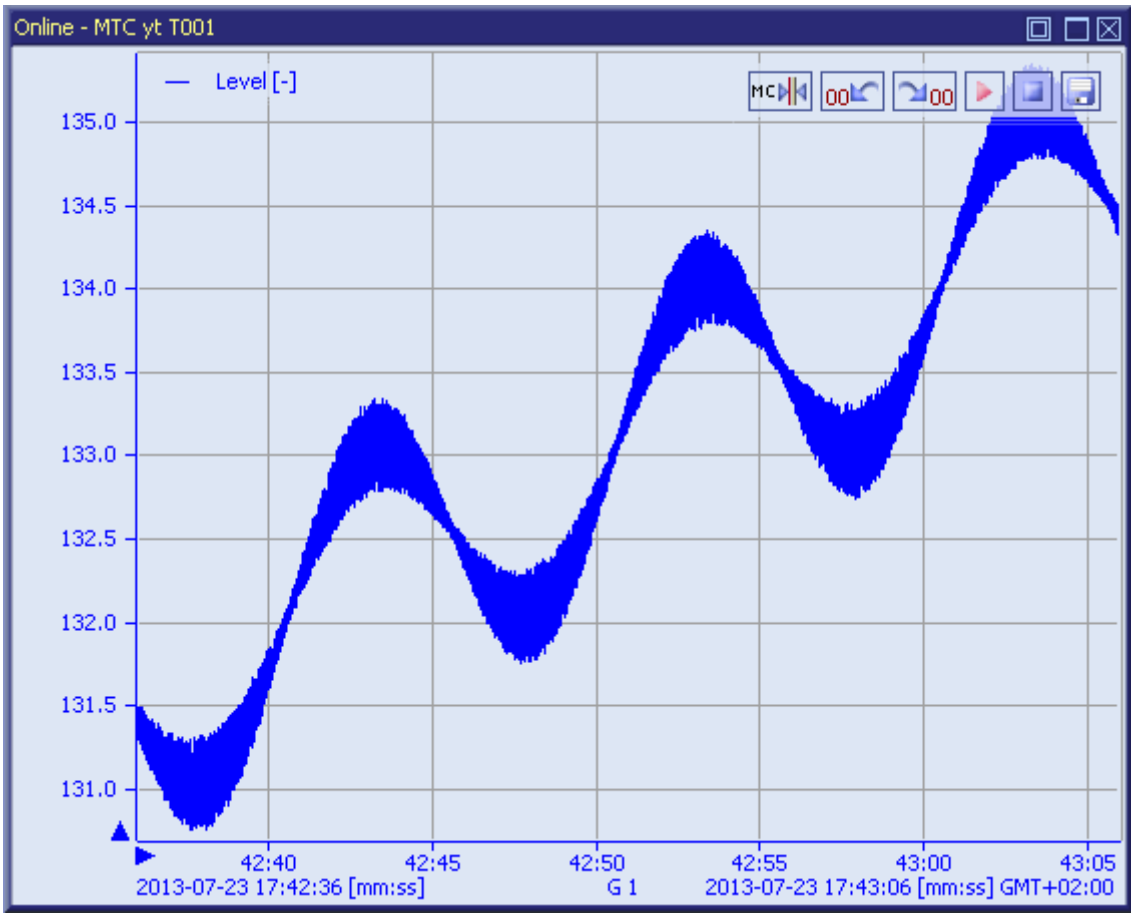


Figure 116: Example of the simulated Data for the File Splitting based Analysis

7.4.11.3 File Splitting based Analysis

The example analysis is found within the Analyzing Script “20 Run File Splitting Processing” and performs the following tasks:

- The online data “Level” is created by the Analyzing Script “10 Create Source Data”.
- The simulated data is stored permanently via the according Storage Profile “Store current Charge”.
- Whenever another file splitting is ready, the created offline data are loaded automatically.
- The Analyzing Model “Create Average” is used in order to determine the average value of the simulated data for the loaded offline data.
- After the average value has been detemined by the first Analyzing Model, the Analyzing Model “Create Trend” is started in order to create an all-time trend of the average value.
- The Monitoring View “Overview” is updated automatically whenever the analysis of another file splitting has completed.

The following box shows the Analyzing Script “20 File Splitting based Processing” from the example location:

```
const ASFName  = "://Measurement Tasks\\File Splitting based Analysis\\10 Create Source Data.asf.xts";
const SPFName  = "://Measurement Tasks\\File Splitting based Analysis\\Permanent Storage.spf.xts";
const AMF1Name = "://Measurement Tasks\\File Splitting based Analysis\\Create Average.amf.xts";
const AMF2Name = "://Measurement Tasks\\File Splitting based Analysis\\Create Trend.amf.xts";


var main()
{
    // start the Analyzing Script which creates our source data
    Start( ASFName, "DoNotWait" );
    Sleep( 5 );

    // start the SPF which does our offline data storing
    Start( SPFName, "DoNotWait" );

    while( 1 )
    {
        if( WaitForFS( SPFName, 90 ) != true )
        {
            // something went wrong, we stop the analysis
            Print( "An unexpected status has been detected, the analysis is stopped." );
            break;
        }

        // here we know that another file splitting has completed
        // load the files from the last file splitting
        LoadData( SPFName,
                  "","Overwrite","","Each Value","",
                  "Load Latest Files");

        // calculate and output the average value of the last file splitting
        Start( AMF1Name, "Wait" );
        Start( AMF2Name, "Wait" );

        Print( "Another file splitting has been processed. Current average: " +
               GetVFrmSD( $'Level (avg, single)', 0 ) );
    }

    // do some clean-up
    Stop( SPFName );
    Stop( ASFName );

}
```

Figure 117: Example of an Analyzing Script for the File Splitting based processing

7.4.11.4 Summary

The following Configuration Files are being used by this chapter:

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\File Splitting based Analysis")

Name	Description
Overview	This Monitoring View displays the simulated data together with the results of the analysis.

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\File Splitting based Analysis")

Name	Description
Create Average	This Analyzing Model calculates the average value of the simulated data from one file splitting.
Create Trend	This Analyzing Model creates an all-time trend of the average values.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\File Splitting based Analysis")

Name	Description
10 Create Source Data	This Analyzing Script creates the simulated data.

20 Run File Splitting Processing	This Analyzing Script performs the actual, file splitting based analysis:
	• the creation of the simulated data is started
	• the permanent storing of the simulated data is started
	• afterwards, the following tasks are running in an endless loop:
	  ○ the Analyzing Script waits until another file splitting completes
	  ○ LoadData() is used in order to load the offline data which was created by the last file splitting
	  ○ the Analyzing Model which analysis the current file splitting is started
	  ○ the Analyzing Model which creates the long-term trend is started

Storage Profiles
(found in "..\Storage Profile Files\Measurement Tasks\File Splitting based Analysis")

Name	Description
Permanent Storage	This Storage Profile constantly stores the simulated data and creates another file splitting each minute.

7.4.12 Passing of Parameters to Analyzing Models

7.4.12.1 General

This chapter shows how to configure an Analyzing Model to accept parameters and how to pass parameter values to the Analyzing Model when it is started via an Analyzing Script. An example use case could be:

- The same analysis shall be performed multiple times in a row, each time with different input and output data.
  - In addition or instead of different input/output data, also any other input argument of Analyzing Functions could be configurable, e.g. the cutoff frequency of a filter, the number of values which shall be collected within a buffer, some threshold, …
- An Analyzing Script can be used in order to start always the same Analyzing Model and to use different parameter values at each call - thus, the one-and-only Analyzing Model could process different data and/or different configurations.

In addition to input arguments of (Analyzing and User) Functions, also the following settings of an Analyzing Model can be used as parameters of an Analyzing Model:

- Analyzing Model Settings table
  - Data Name Prefix of Input Data
  - Data Name Prefix of Output Data
- Analyzing Model Parameters table
  - Cycle Time/Rate
- Offline Time Configuration table
  - Time Input Mode
  - Minimum Time Format
  - Minimum Time Value
  - Time Interval Format
  - Time Interval Value
  - Maximumt Time Format
  - Maximum Time Value

7.4.12.2 Analyzing Models

In order to add parameters to an Analyzing Model, the Add Parameter context menu item must be chosen. This item is available from the context menu which is called above any input argument of an Analyzing Function or User Function within the **AME Standard T001** and from several settings within the present tables (see above). Only input arguments which are represented through simple data types (Boolean, Integer, Decimal and String) can be used as parameters of an Analyzing Model.

All of the currently defined parameters of the Analyzing Model are listed within the Provided Parameters table. Within this table, the names, unit, description and default value of each parameter as well as the order of parameters can be configured.

The following screenshot shows an example of a Provided Parameters table which contains some definitions already. It has been taken from the example Analyzing Model “Analyzing Model with Parameters”:

Provided Parameters

No.	Full Name	Short Name	Unit	Description	Default Value	Parent Function	Parent Argument
1	Shape	Shape	n/a	form of the wave which shall be generated	Sine	GenWF	Shape
2	Amplitude	Amp	1	amplitude of the waveform which shall be generated	10	GenWF	Amp
3	Frequency	Frq	Hz	frequency of the waveform which shall be generated	0.5	GenWF	Frq
4	Offset	Off	1	steady component of the waveform which shall be generated	2.5	GenWF	Off
5	Phase	Pha	rad	phase-shift of the waveform which shall be generated	0	GenWF	Pha
6	Name	Name	n/a	name of the data or values which shall be written to the system	Default Name	Out	Name

Figure 118: Example of a Provided Parameters Table

7.4.12.3 StartEx()

In case an Analyzing Model is started without passing of any parameters to it (e.g. via the **ANS Explorer** or via the Analyzing Function **Start()**), it is initialized with the default values from the Provided Parameters table.

Up to 128 parameters can be configured for each Analyzing Model. In order to pass parameter values to an Analyzing Model during its start, the Analyzing Model must be started via the Analyzing Function **StartEx()** - thus, starting of Analyzing Models with parameters must be performed via an according Analyzing Script. The order of parameters which are passed to **StartEx()** (left to right) must be identical to the order of parameters within the Provided Parameters table (top to bottom).

The examples below show different ways to call an Analyzing Model which provides configurable parameters. All Analyzing Scripts from below are found also in “..\Analyzing Script Files\Measurement Tasks\Passing of Parameters”:

```
var main()
{
    // the following command calls the Analyzing Model without passing of any parameters to it -
    // as a result, the Analyzing Model will run with its configured default parameter values
    Start("://Measurement Tasks/Passing of Parameters/Analyzing Model with Parameters.amf.xts",
          "Wait");

    // Note: the Analyzing Model is configured to exit automatically after 10 seconds of operation
}
```

Figure 119: Example of an Analyzing Script which does not pass any Parameters

```
var main()
{
    // the following command calls the Analyzing Model with passing of a few parameters to it -
    // as a result, the Analyzing Model will use the provided values for the provided parameters
    // and the default parameter values for the not-provided parameters
    StartEx("://Measurement Tasks/Passing of Parameters/Analyzing Model with Parameters.amf.xts",
            "Wait", /*Timeout=*/ 0, /*Shape=*/ "Rectangle", /*Amplitude=*/ 100; /*Enable=*/ true);

    // Note: the Analyzing Model is configured to exit automatically after 10 seconds of operation
}
```

Figure 120: Example of an Analyzing Script which passes a few Parameters

```
var main()
{
    // the following command calls the Analyzing Model with passing of all parameters to it -
    // as a result, the Analyzing Model will use the provided values for all of its parameters
    StartEx("://Measurement Tasks/Passing of Parameters/Analyzing Model with Parameters.amf.xts",
            "Wait", /*Timeout=*/ 0,
            /*Shape=*/ "Sawtooth", /*Amplitude=*/ 100, /*Frequency=*/ 2,
            /*Offet=*/ 0, /*Phase=*/ 0, /*Name=*/ "Another Name";);

    // Note: the Analyzing Model is configured to exit automatically after 10 seconds of operation
}
```

Figure 121: Example of an Analyzing Script which passes all Parameters

7.4.12.4 Summary

The following Configuration Files are being used by this chapter:

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Passing of Parameters")

Name	Description
Analyzing Model with Parameters	This Analyzing Model provided parameters that can be used in order to configure the shape and the name of the created output data.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Passing of Parameters")

Name	Description
10 Calling without passing of Parameters	This Analyzing Script calls the above Analyzing Model without passing of any parameters to it.
20 Calling with passing of a few Parameters	This Analyzing Script calls the above Analyzing Model and passes a few parameters to it.
30 Calling with passing of all Parameters	This Analyzing Script calls the above Analyzing Model and passes all parameters to it.

7.4.13 Extrapolation

7.4.13.1 General

This chapter shows how to use the Analyzing Function **Extrapolate()** in order to extrapolate the known trend of some data into the future. An example use case could be:

- There is an analysis which creates a trend of some key value, e.g. a maximal temperature or a maximal amplitude of a vibration.
- After the trend has been created for some time, it shall be found out at which moment in future a certain threshold will be violated.

7.4.13.2 Extrapolate()

The following example shows multiple calls of **Extrapolate()**, each one with a different order. It is equivalent to the Analyzing Script “20 Extrapolation”:

```
var main()
{
    // order = 3
    Extrplt( $'Ramp', "Ramp Extrapolated, 03. Order", "Decimal32", 3, 86400, 1, true );
    Print( "order 3: complete" );

    // order = 4
    Extrplt( $'Ramp', "Ramp Extrapolated, 04. Order", "Decimal32", 4, 86400, 1, true );
    Print( "order 4: complete" );

    // order = 5
    Extrplt( $'Ramp', "Ramp Extrapolated, 05. Order", "Decimal32", 5, 86400, 1, true );
    Print( "order 5: complete" );

}
```

*Figure 122: Example of an Analyzing Script which uses **Extrapolate()***

In order to get the input data (“Ramp”), load this data either from the “..\X-Tools Offline Data Files\Measurement Tasks\Extrapolation” folder or start the Analyzing Script “10 Generation”.

The extrapolated data appears as offline data within the **MDS Explorer** as soon as the calculation has finished. The following screenshot shows a visualization of both the input and the extrapolated data:

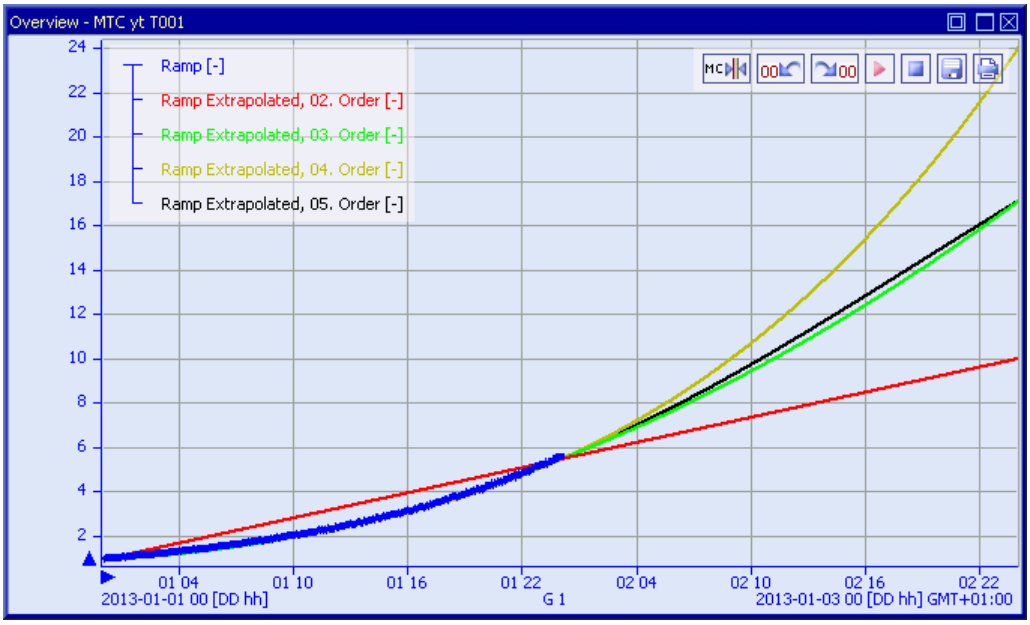


*Figure 123: Example Output of **Extrapolate()***

7.4.13.3 Detection of Violation

After the extrapolated data has been calculated, another Analyzing Script can be used in order to check whether the extrapolated data violates a given threshold at any moment. The following Analyzing Script is identical to “30 Detection of Violation” and outputs the data “Violation” in case a violation is detected. The one-and-only value of the data “Violation” is placed at the time when the first violation is detected:

```
var Threshhold = 15;
var Violation  = false;


var main()
{
    if( Violation )
    {
        // we have found the first violation already, there is nothing else to do
    }
    else
    {
        // we did not detect a violation yet
        if( $'Ramp Extrapolated, 04. Order' >= Threshhold )
        {
            // we have found the first violation
            Violation = true;
            Print( "A violation has been detected." );

            Out( "Violation", $'Ramp Extrapolated, 04. Order', true, "", "",
                 "Decimal32", "Default", "Hysteresis = 0.000; Interval = 1000000000.",
                 "Relative Size", 100, "Single Value" );
        }
    }

}
```

Figure 124: Example of an Analyzing Script which searches for a Violation of the given Threshhold

The following screenshot shows the input data, the extrapolated data and the violation which has been detected by the above Analyzing Script:

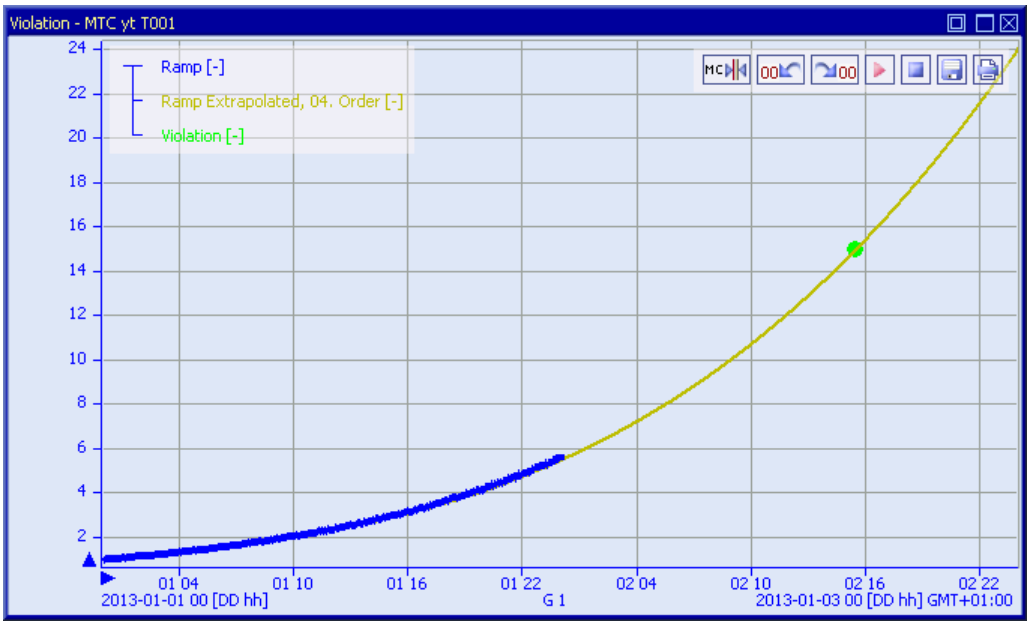


Figure 125: Extrapolated Data together with the detected Violation

7.4.13.4 Summary

The following Configuration Files are being used by this chapter:

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\Extrapolation")

Name	Description
Overview	Shows the input data together with the results of the extrapolation and with the found violation.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Extrapolation")

Name	Description
10 Generation	This Analyzing Script creates the input data which represents an example trend.
20 Extrapolation	This Analyzing Script calculates the extrapolation of the example trend into the future.
30 Detection of Violation	This Analyzing Script checks whether the extrapolated values violated the given threshold at any moment in future.

7.4.14 Data Analysis with Gaps

7.4.14.1 General

This chapter shows how to consider gaps within offline data during the analysis via Analyzing Models and/or Analyzing Scripts. Gaps within the present offline data can occur because the data source was not constantly available, because the offline data was stored with some start- and stop triggers (= the storing was not active constantly) or because only parts of the offline data files have been loaded from the disk.

In all cases, there may be a need to put special attention onto the fact that the offline data contains periods of time for which there is no data available.

The following examples show how to calculate the average value for offline data which contains gaps.

7.4.14.2 Action based Calculation Mode

The Analyzing Model "10 Action based Calculation" shows how to use the action based calculation mode in order to perform the calculation only at the moments at which there is a value available for the input data. The Analyzing Model itself is rather simple, it accumulates all temperature values and divides them by the so-far processed cycles:

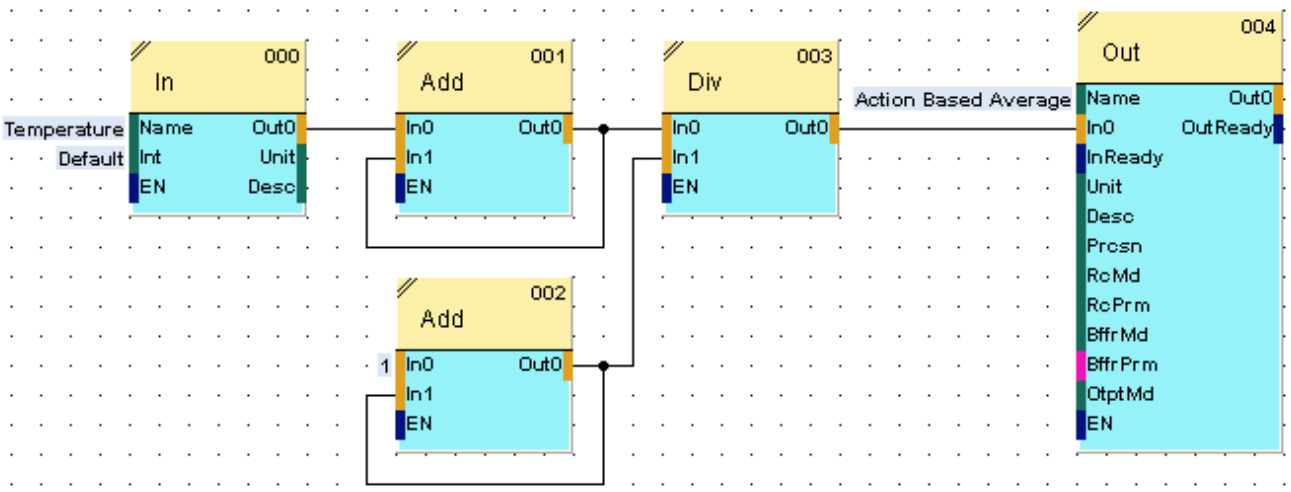


Figure 126: Analyzing Model which uses the Action based Calculation Mode for calculating of an Average

The following screenshot shows the input and output of this Analyzing Model. The input data can be loaded from "..\X-Tools Offline Data Files\Data Analysis with Gaps":

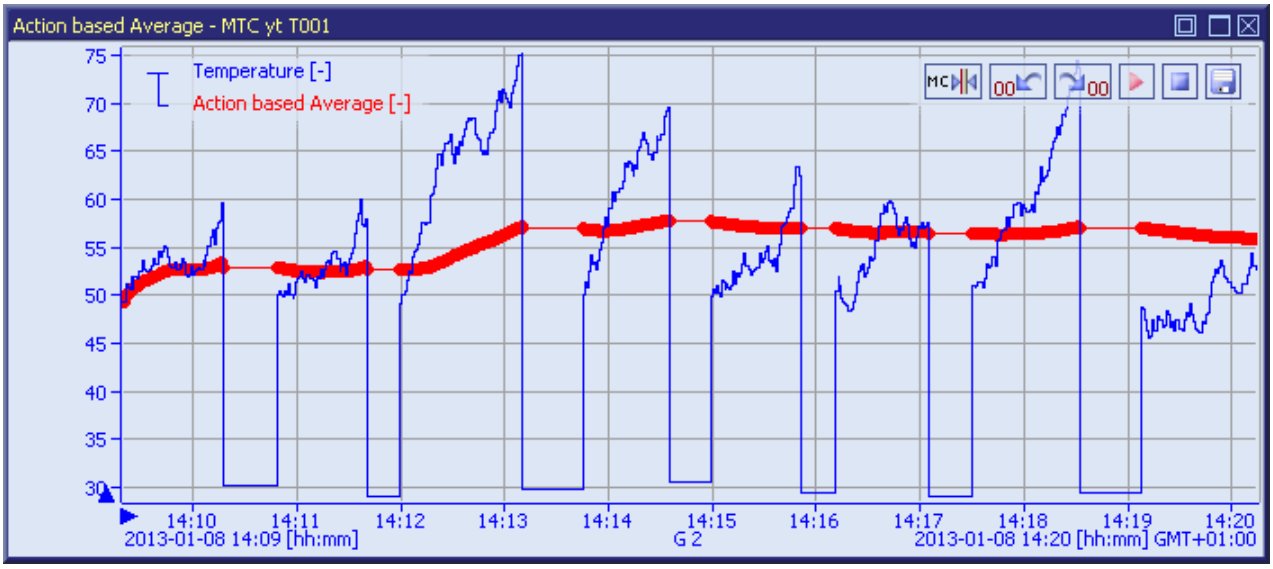


Figure 127: Example of an Average which was calculated via the Action based Calculation Mode

It can be seen that the output does not contain values at moments where there are no input values available. In addition, after each gap the average value contains at the last known value before the gap (thus, the average value is not falsified during the period of time at which the input data does not contain values).

7.4.14.3 Falsified Average

The Analyzing Model "20 Falsified Average" is identical to "10 Action based Calculations" but it uses a different configuration within the Analyzing Model Parameters table:

- The Calculation Mode is "Equidistant" (and not "Action based and once")

This different configuration is needed because the output data of the Analyzing Model shall be equidistant now (remember that equidistant data can be required by different successive analysis). The following screenshot shows the input and output of this Analyzing Model:

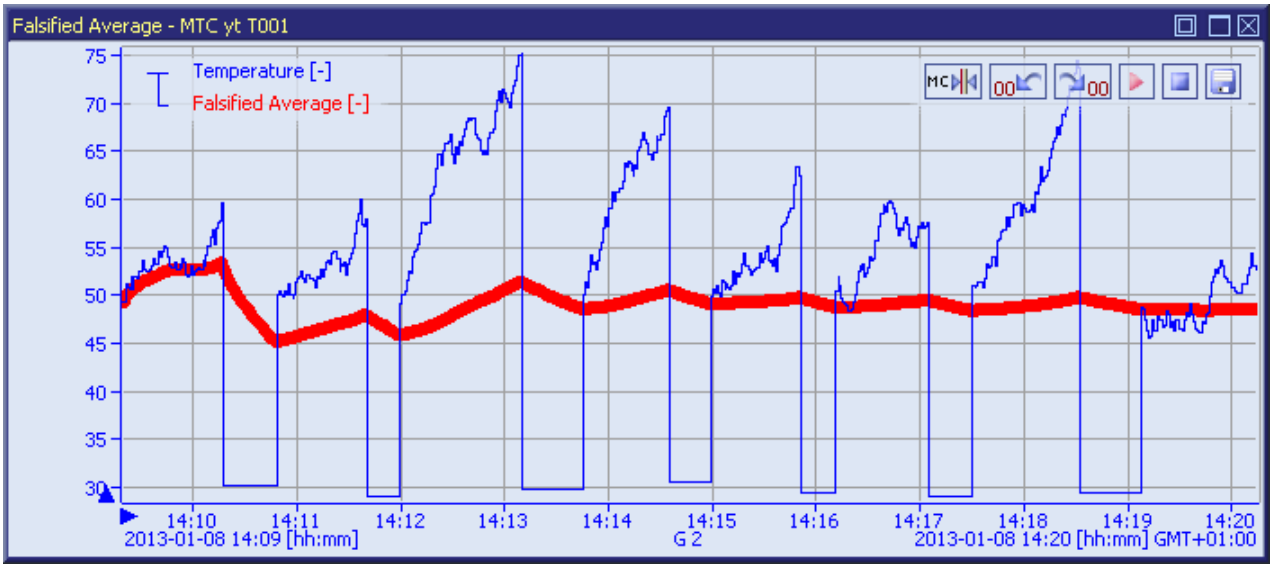


Figure 128: Example of a falsified Average

It can be seen that the output contains values also at the moments where there are no input values available - this is basically OK and wanted in this way. However, the Analyzing Model continues to calculate the average on base of the last-known value also for the periods of time where there are no input values available, and as a result the whole average is falsified from the moment of the first gap.

7.4.14.4 True Average

The Analyzing Model “30 True Average 1” shows how both needs can be achieved: a true average and equidistant output data. It looks like this:

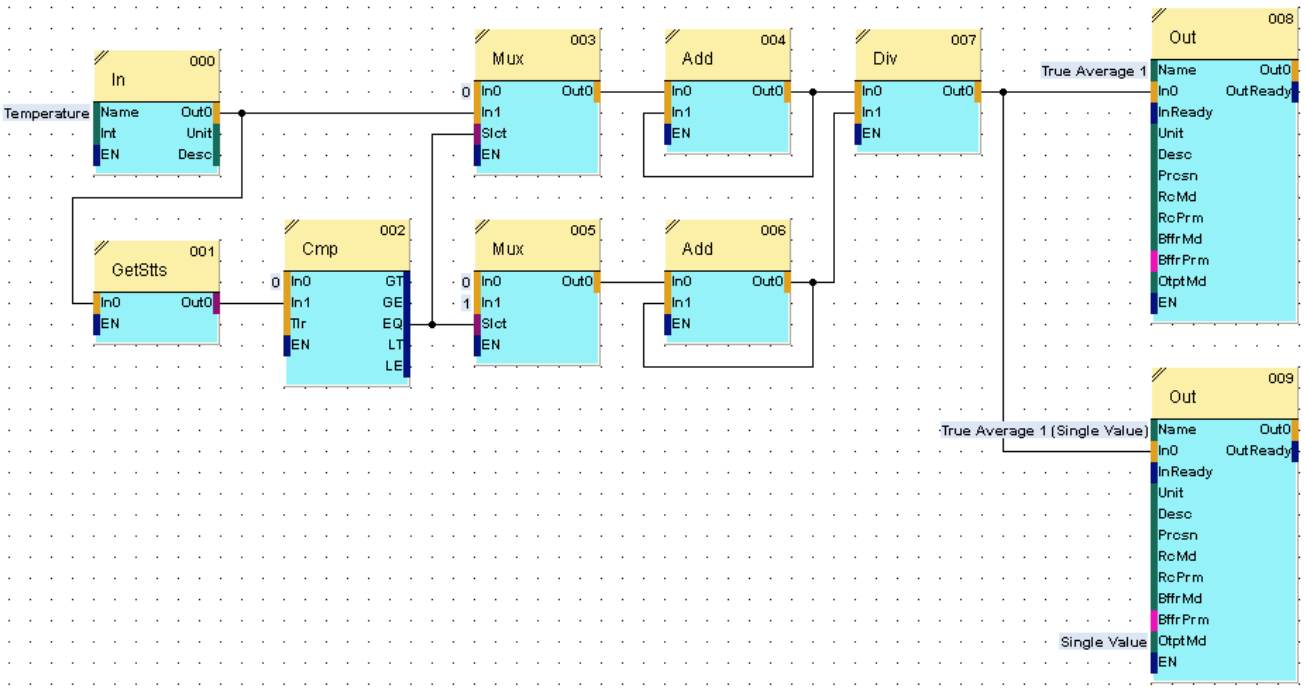


Figure 129: Analyzing Model which uses GetStatus() in order to be able to determine the true Average

The Analyzing Function **GetStatus()** is used by the Analyzing Model in order to check whether the output of the Analyzing Function **Input()** provides valid data at the moment or not. In case the data is valid, **GetStatus()** returns 0. In all other cases, it returns the according status value. The construct of **GetStatus()**, **Compare()** and the two **Multiplexer()** ensure that the average value is updated only at moments when the input data provides a valid value. In case the input data is not valid, the sum and the value counter are not increased.

The following screenshot shows the input and output of this Analyzing Model:

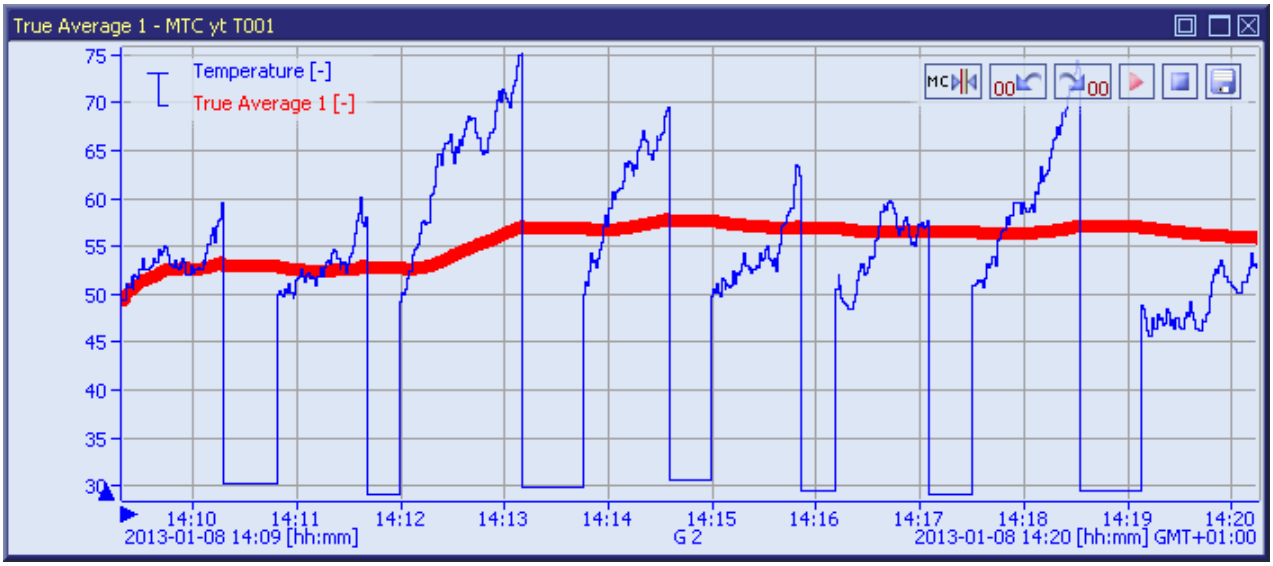


Figure 130: Example of an Average which was calculated with considering of Gaps

It can be seen that the output contains values also at the moments where there are no input values available. In addition, the calculated average values are identical to the ones from the action based calculation. While the input data does not provide values, the average value does not change.

The Analyzing Model “40 True Average 2” is identical to the above Analyzing Model but it uses the “Enable” input argument of the **Output()** Analyzing Function in order to disable the output whenever the input data does not provide valid values. The output of this Analyzing Model is identical to the output from the action based Analyzing Model from above and shows that different attempts can be used in order to achieve identical results. Depending to the actual application, the one or the other approach can be more suitable (for example, Analyzing Models with more than one input data may not work properly in action based calculation modes in case the input data have values at different moments in time).

7.4.14.5 Automatic Analysis

There are also applications which require to perform repeated analysis for defined time slices, e.g. to calculate one average value per day on base of the data from the whole day. Also for such applications it can happen that there are gaps within the to-be-analysed offline data.

The Analyzing Script “20 Automatic Analysis” shows an example for such an automatic analysis. It performs the following tasks:

- At the beginning, it loads all of the to-be-processed offline data in order to determine the first and last available timestamps.
- After the first and last available timestamps are known, it divides the present offline data into chunks of equal time (in this case, 30 seconds).
- For each chunk of data, it loads the data values from the according interval of time and it calculates the average value for this chunk.
- When the average value for the current chunk is known, it is added to the trend of average values. Thus, there is one average value each 30 seconds.
    - There is also a noticeable exception: In case there is no offline data file which covers the start time which is passed to **LoadData()**, the loaded offline data starts after the desired start time (in such a case, it starts a the first point in time after the specified start time which is covered by an offline data file). Accordingly, the loaded offline data also stops before the desired stop time in case there is no offline data file which covers the specified stop time (in such case, it stops at the last point in time before the specified stop time which is covered by an offline data file).

The Monitoring View “Automatic Analysis” can be used in order to see the calculated trend of average values after the calculation has completed.

7.4.14.6 Summary

The following Configuration Files are being used by this chapter:

Monitoring Views
(found in “..\Monitoring View Files\Measurement Tasks\Data Analysis with Gaps”)

Name	Description
Automatic Analysis	Shows the results from the automatic, interval-based analysis of all offline data.
Average	Shows the different results (average values) from the example Analyzing Models.

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Data Analysis with Gaps")

Name	Description
10 Action based Calculation	This Analyzing Model uses the action-based operating mode in order to calculate an all-time average for the provided input data.
20 Falsified Average	This Analyzing Model shows how a falsified average could be created in case gaps are not considered properly.
30 True Average 1	This Analyzing Model shows how to consider gaps properly in order to receive the proper average and outputs equidistant data.
40 True Average 2	This Analyzing Model shows how to consider gaps properly in order to receive the proper average and outputs data only for time intervals at which the input data is available.
50 Create Trend 1	This Analyzing Model creates a trend with a single value each defined interval.
60 Create Trend 2	This Analyzing Model creates a trend with all values from each processed interval.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Data Analysis with Gaps")

Name	Description
10 Generation	This Analyzing Script in combination with the below Storage Profile can be used in order to create offline data files with variable length and with a variable time without storing in between two offline data files.
20 Automatic Analysis	This Analyzing Script runs all of the tasks which are needed for an automatic analysis of all available offline data.

Storage Profiles
(found in "..\Storage Profile Files\Measurement Tasks\Data Analysis with Gaps")

Name	Description
Offline Data with Gaps	This Storage Profile uses the ANS-generated start and stop triggers in order to create offline data files with variable length and with a variable time without storing in between two offline data files.

7.4.15 Output Modes of Output Analyzing Functions

7.4.15.1 General

This chapter shows how to use the “OutputMode” setting of output Analyzing Functions in order to handle different application scenarios. The “OutputMode” setting can be used for calculations which output offline data and allows to choose how the calculated results shall be added to the system - the possible options are:

- Overwrite
  - In case an offline data with specified name exists already, it is overwritten at the end of the calculation and receives the values from the current calculation. All previous values of the specified offline data disappear at the moment of the overwrite.
- Append
  - In case an offline data with specified name exists already, the values from the current calculation are appended to the already present values of the offline data.
  - In case the stop time of the already present offline data is after or equal to the start time of the to-be-appended values from the current calculation, the append fails and the parent Analyzing Program suspends.
- Advanced Append
  - Basically, this mode works like the simple “Append”.
  - In addition, the “BufferMode” and “BufferParameter” settings of the Analyzing Function can be used in order to specify the maximal number of values which shall be allowed. In case the append would result in a bigger output data, the oldest values are removed from the offline data automatically.
- Single Value
  - The created offline data contains exactly one value.
  - In case multiple values are written to the output data during the processing of the parent Analyzing Program, the probe from the last write always overwrites the probe from all previous writes.

7.4.15.2 Analyzing Model

The Analyzing Model “Output Modes” generates a simple ramp and outputs this ramp via four different **Output()** Analyzing Functions. Each Analyzing Function uses a different “OutputMode”:

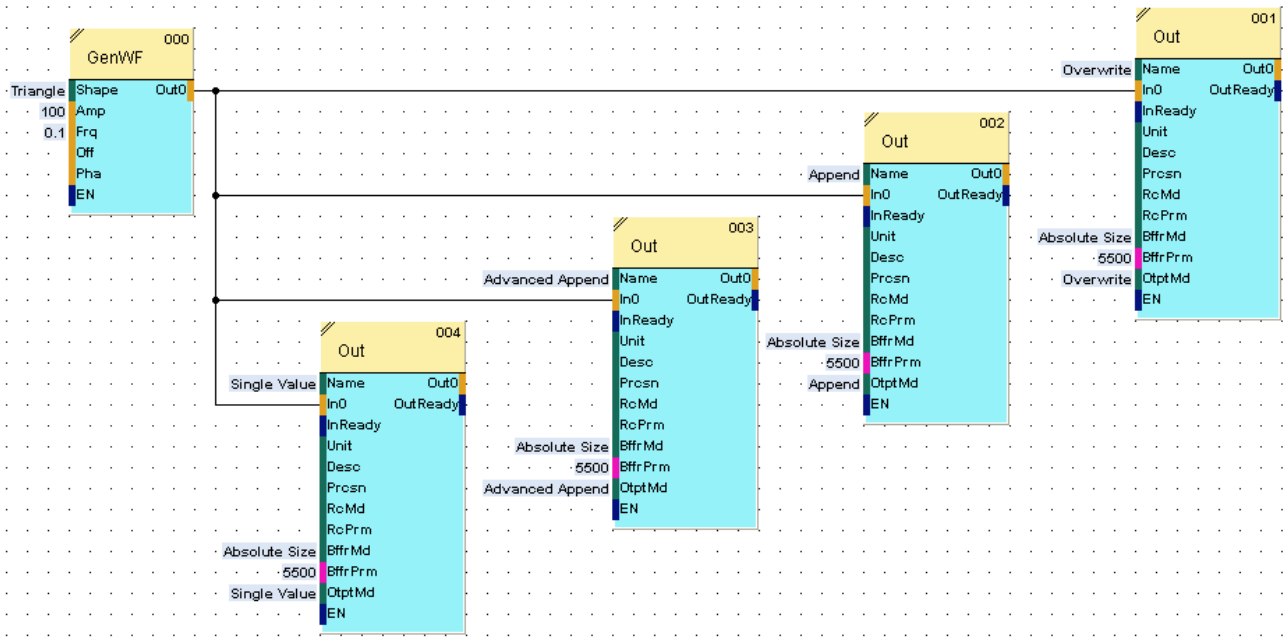


Figure 131: Analyzing Model which uses different Output Modes

Run this Analyzing Model at least 6 times. It is configured to create 1 second of output data, thus it stops automatically 1 second after each start. After each run, check the number of values which are hold by the offline data from the **MDS Explorer**. The following table shows the expected number of values after each run:

Run	Overwrite	Append	Advanced Append	Single Value
1	1001	1001	1001	1
2	1001	2002	2002	1
3	1001	3003	3003	1
4	1001	4004	4004	1
5	1001	5005	5005	1
6	1001	6006	5500	1

As expected, the created offline data behave as follows:

- The offline data “Overwrite” contains always 1001 values. After each run of the Analyzing Model, these 1001 values represent the values from this run.
- The offline data “Append” receives additional 1001 values after each run of the Analyzing Model.
- The offline data “Advanced Append” receives additional 1001 values after each run of the Analyzing Model until the specified buffer size (5500 values) is reached - from this moment on, the oldest values are removed and the number of values does not grow above 5500 values.
- The offline data “Single Value” contains always 1 value. After each run of the Analyzing Model, this 1 value contains the last value from this run.
- All output modes except “Advanced Append” ignore the configured buffer parameters.

The Monitoring View “Output Modes” also can be used in order to take a look onto all four output data. Within the Monitoring View, the different behavior of the different output modes is clearly visible.

7.4.15.3 Analyzing Script

The Analyzing Script “Output Modes” produces exactly the same result as the above Analyzing Model. It looks like this:

```
var main()
{
    var Signal;


    Signal = GenWF( "Triangle", 100, 0.1 );


    Out( "Overwrite",       Signal, true, "", "", "Decimal64", "Default",
         "Hysteresis = 0.000; Interval = 1000000000.", "Absolute Size", 5500, "Overwrite" );
    Out( "Append",          Signal, true, "", "", "Decimal64", "Default",
         "Hysteresis = 0.000; Interval = 1000000000.", "Absolute Size", 5500, "Append" );
    Out( "Advanced Append", Signal, true, "", "", "Decimal64", "Default",
         "Hysteresis = 0.000; Interval = 1000000000.", "Absolute Size", 5500, "Advanced Append" );
    Out( "Single Value",    Signal, true, "", "", "Decimal64", "Default",
         "Hysteresis = 0.000; Interval = 1000000000.", "Absolute Size", 5500, "Single Value" );


}
```

Figure 132: Analyzing Script which uses different Output Modes

7.4.15.4 Summary

The following Configuration Files are being used by this chapter:

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\Output Modes")

Name	Description
Output Modes	Shows the values of all four offline data and updates the displayed time interval automatically after each run of the Analyzing Model.

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Output Modes")

Name	Description
Output Modes	This Analyzing Model uses all four output modes in order to show their different behavior.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Output Modes")

Name	Description
Output Modes	This Analyzing Script uses all four output modes in order to show their different behavior.

7.4.16 #include within Analyzing Scripts

7.4.16.1 General

This chapter shows how to use #include statements within Analyzing Scripts. #include statements can be used in order to include variables, constants and functions which are present in any other Analyzing Script. An example use case could be:

- Multiple Analyzing Scripts share the same variables, constants and or functions.
- When all of this common functionality is provided by some central Analyzing Scripts, the main Analyzing Scripts can include these central ones and can use the contained variables, constants and functions as if they were defined directly within each of these main Analyzing Scripts.

7.4.16.2 Preparing of Analyzing Scripts which shall be #included later

The following examples show some simple Analyzing Scripts which provide common/basic functionality. Note that an Analyzing Script which is only used to be #included later does not require any function, even not the main() function:

```
const MY_TRUE  = 1;
const MY_FALSE = 0;
const MY_NULL  = 0;
```

Figure 133: The Analyzing Script "05 Defines" from the Example Location

```
#include "://Measurement Tasks/#include within Analyzing Scripts/05 Defines.asf.xts"


var MyFunction1( var V1, var V2 )
{
    return V1 * V2;

} // MyFunction1


var MyFunction2( var V1, var V2 )
{
    return V1 + V2;

} // MyFunction2
```

```
var MyFunction3( var V1, var V2 )
{
    if( V2 == MY_NULL )
    {
        // do not allow a division by zero
        return 0;
    }

    return V1 / V2;

} // MyFunction3
```

Figure 134: The Analyzing Script "10 Functions" from the Example Location

```
#include "://Measurement Tasks/#include within Analyzing Scripts/05 Defines.asf.xts"


const    MyConst1 = 3.1415927;
const    MyConst2 = 2.7182818;
var      MyVar1   = 2005;
var      MyVar2   = 2012;
```

Figure 135: The Analyzing Script "10 Types" from the Example Location

7.4.16.3 #including of Analyzing Scripts

The following example shows a sequential Analyzing Script which uses the variables, constants and functions of the #included Analyzing Scripts:

```
#include "://Measurement Tasks/#include within Analyzing Scripts/10 Types.asf.xts"
#include "://Measurement Tasks/#include within Analyzing Scripts/10 Functions.asf.xts"


var main()
{
    Print( "MyConst1 = " + MyConst1 );
    Print( "MyConst2 = " + MyConst2 );

    MyVar1 -= 505;
    MyVar2 -= 12;

    Print( "MyVar1 = " + MyVar1 );
    Print( "MyVar2 = " + MyVar2 );

    Print( MyVar1 + " * " + MyVar2 + " = " + MyFunction1( MyVar1, MyVar2 ) );
    Print( MyVar1 + " + " + MyVar2 + " = " + MyFunction2( MyVar1, MyVar2 ) );

    Print( MyVar1 + " / " + MyVar2 + " = " + MyFunction3( MyVar1, MyVar2 ) );

    // try to provoke a division by zero
    MyVar2 = 0;
    Print( MyVar1 + " / " + MyVar2 + " = " + MyFunction3( MyVar1, MyVar2 ) );

}
```

Figure 136: Example of a sequential Analyzing Script which uses #include Statements

The above Analyzing Script creates an output like in the following screenshot:

```
MyConst1 = 3.14159269
MyConst2 = 2.7182818
MyVar1 = 1500
MyVar2 = 2000
1500 * 2000 = 3000000
1500 + 2000 = 3500
1500 / 2000 = 0.75
1500 / 0 = 0
```

Figure 137: Output of a sequential Analyzing Script which uses #include Statements

Onother example shows a time slice based Analyzing Script which outputs online data and which uses the variables and constants of the #included Analyzing Script:

```
#include "://Measurement Tasks/#include within Analyzing Scripts/10 Types.asf.xts"


var main()
{
    ~'Type Test 1' = MyConst1;
    ~'Type Test 2' = MyConst2;

    ~'Type Test 3' = MyVar1;
    ~'Type Test 4' = MyVar2;

    MyVar1 += 1;
    MyVar2 /= 1.005;

}
```

Figure 138: Example of a Time Slice based Analyzing Script which uses #include Statements

The online data which is created by the above Analyzing Script is shown within the following screenshot:

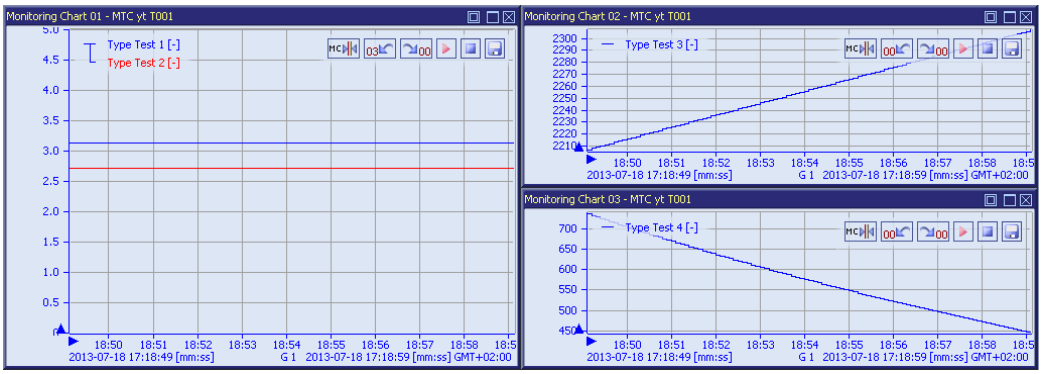

Monitoring Chart 01 - MTC yt T001
Type Test 1 [-]
Type Test 2 [-]
Monitoring Chart 02 - MTC yt T001
Type Test 3 [-]
Monitoring Chart 03 - MTC yt T001
Type Test 4 [-]
2013-07-18 17:18:49 [mm:ss]
2013-07-18 17:18:59 [mm:ss] GMT+02:00

Figure 139: Output of a Time Slice based Analyzing Script which uses #include Statements

7.4.16.4 Summary

The following Configuration Files are being used by this chapter:

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\#include within Analyzing Scripts")

Name	Description
20 Time Slice based	This Monitoring View shows the data which is output by the Analyzing Script "20 Time Slice based".

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\#include within Analyzing Scripts")

Name	Description
05 Defines	This Analyzing Script contains basic defines which are included by the other Analyzing Scripts from this measurement task.
10 Functions	This Analyzing Script contains simple functions which are included by other Analyzing Scripts from this measurement task.
10 Types	This Analyzing Script contains simple varibles and constants which are included by other Analyzing Scripts from this measurement task.
20 Sequential	This Analyzing Script uses the #include statement in order to access the variables, constants and functions from other Analyzing Scripts.
20 Time Slice based	This Analyzing Script outputs online data and uses the #include statement in order to access the variables and constants from other Analyzing Scripts.

7.4.17 InConst() and Single Point Parameter Definitions

7.4.17.1 General

There are several Analyzing Functions with input arguments which can not change during runtime (e.g. the number of points of Buffer(), …). These input arguments are of class “parameter” and can not be connected to a link. However, for Analyzing Models and User Functions, it sometimes may be good to have one place which configures multiple input arguments of class “parameter” simultanesouly (e.g. the number of points of multiple Buffer() Analyzing Functions, …).

For this purpose, the Analyzing Function InConst() is available within Analyzing Models and User Functions. In addition, the “const” statement can be used within Analyzing Scripts for the same purpose.

7.4.17.2 User Functions

The following example shows a User Function which uses InConst() in order to configure the “parameter” input arguments of multiple Analyzing Functions. In addition, the input of InConst() is added as input interface of the User Function so that its value also can be configured through the calling Analyzing Model/Analyzing Script:

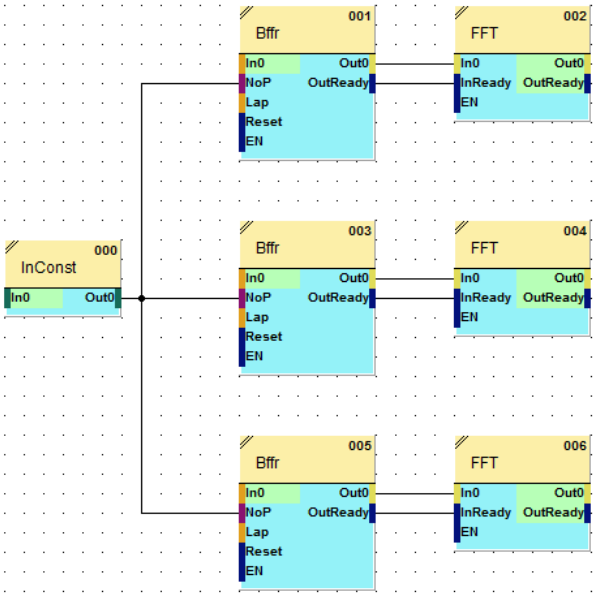


Figure 140: The User Function “Example for InConst()” from the Example Location

The above User Function can be called by the following Analyzing Model:

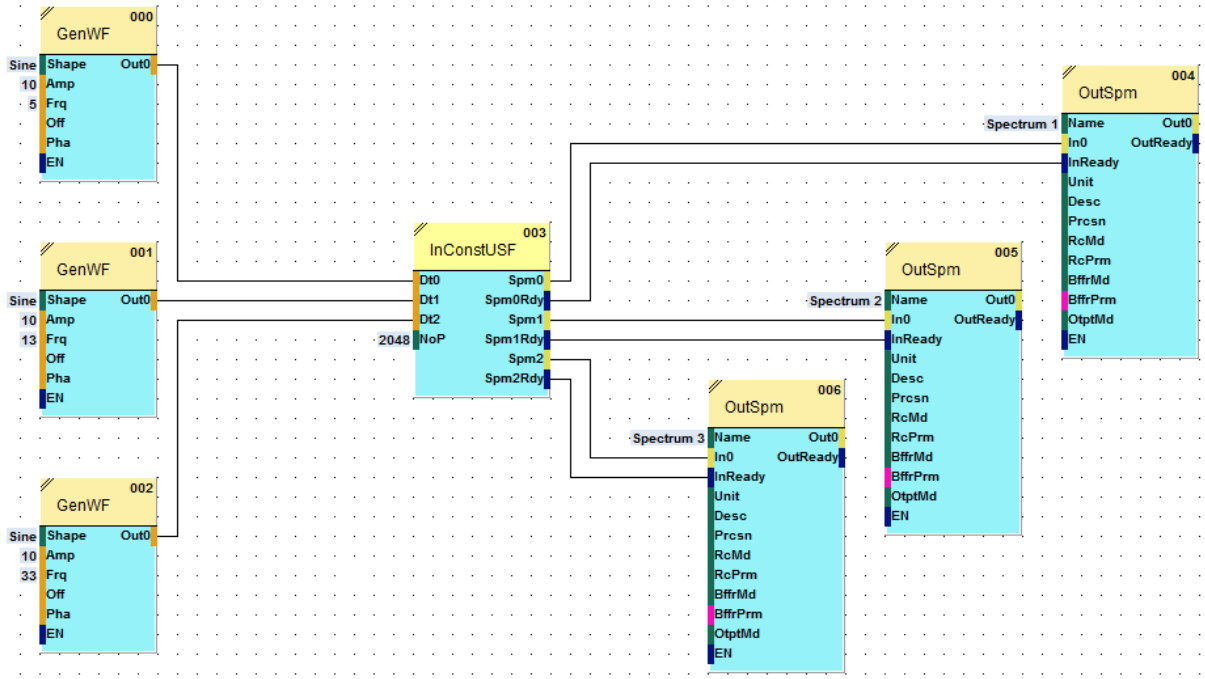


Figure 141: The Analyzing Model "Example with User Function" from the Example Location

An example of the output of the Analyzing Model is shown by the following screenshot:

Figure 142: Example output of the Monitoring View "Example with User Function"

7.4.17.3 Analyzing Models

The following example shows an Analyzing Model which uses InConst() in order to configure the “parameter” input arguments of multiple Analyzing Functions. In addition, the input of InConst() is added as parameter of the Analyzing Model so that its value also can be configured through a calling Analyzing Script:

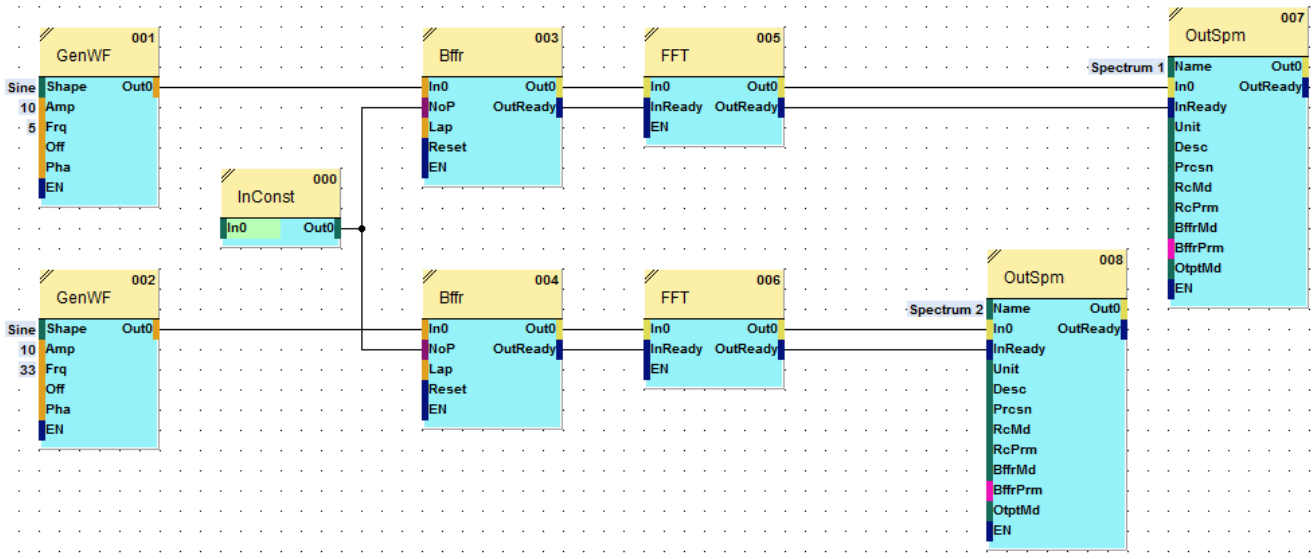


Figure 143: The Analyzing Model “Example for InConst()” from the Example Location

The above Analyzing Model can be started directly. In addition, it also can be called by the following Analyzing Script:

```
var main()
{
    StartEx( "://Measurement Tasks/InConst()/Example for InConst().amf.xts", "Wait", 0, 2048; );

}
```

Figure 144: The Analyzing Script “Example with Analyzing Model” from the Example Location

An example of the output of the Analyzing Script is shown by the following screenshot:

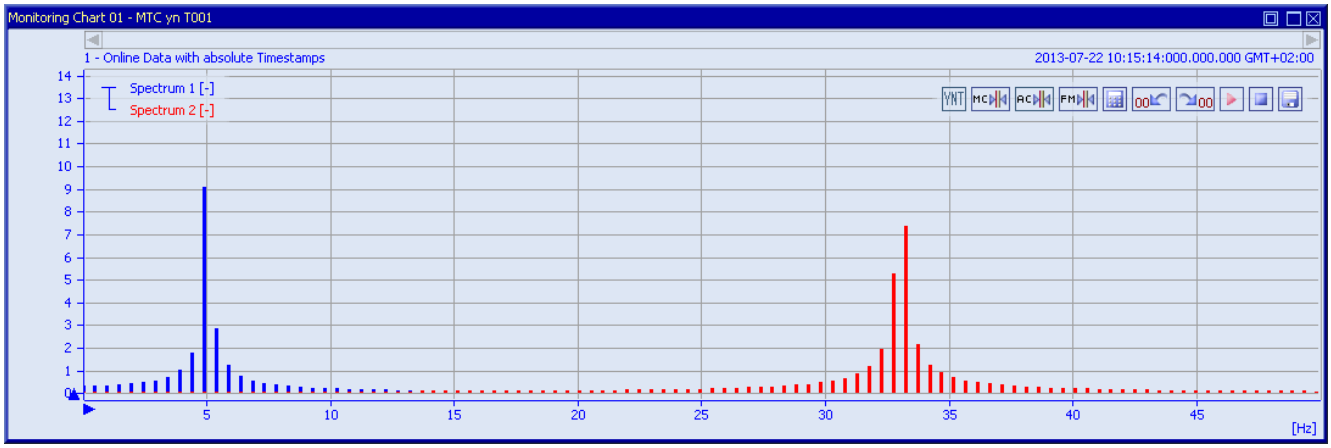

Monitoring Chart 01 - MTC yn T001
1 - Online Data with absolute Timestamps
2013-07-22 10:15:14:000.000.000 GMT+02:00
Spectrum 1 [-]
Spectrum 2 [-]
[Hz]

Figure 145: Example output of the Monitoring View “Example for InConst()” from the Example Location

7.4.17.4 Analyzing Scripts

The following example shows an Analyzing Script which uses the “const” statement in order to configure the “parameter” input arguments of multiple Analyzing Functions:

```
const NoP = 2048;


var main()
{
    var Buffer;
    var BufferReady;
    var Spectrum;
    var SpectrumReady;
    var Data0 = GenWF( "Sine", 10, 100 );
    var Data1 = GenWF( "Sine", 10, 330 );

    Buffer   = Bffr( Data0, NoP, 0, false, true, BufferReady );
    Spectrum = FFT( Buffer, BufferReady, true, SpectrumReady );
    OutSpm( "Spectrum 1", Spectrum, SpectrumReady );

    Buffer   = Bffr( Data1, NoP, 0, false, true, BufferReady );
    Spectrum = FFT( Buffer, BufferReady, true, SpectrumReady );
    OutSpm( "Spectrum 2", Spectrum, SpectrumReady );

}
```

Figure 146: The Analyzing Script “Example for const” from the Example Location

An example of the output of the Analyzing Script is shown by the following screenshot:

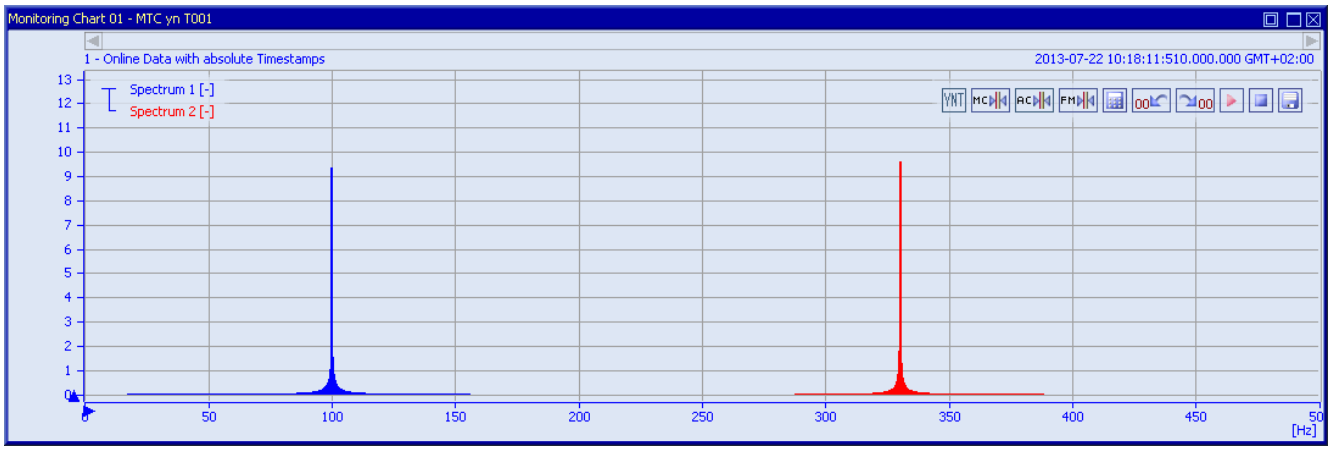


Figure 147: Example output of the Monitoring View “Example for const” from the Example Location

7.4.17.5 Summary

The following Configuration Files are being used by this chapter:

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\InConst()")

Name	Description
Example for const	This Monitoring View shows the data which is output by the Analyzing Script "Example for const".
Example for InConst()	This Monitoring View shows the data which is output by the Analyzing Model "Example for InConst()" and by the Analyzing Script "Example with Analyzing Model".
Example with User Function	This Monitoring View shows the data which is output by the Analyzing Model "Example with User Function".

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\InConst()")

Name	Description
Example for InConst()	This Analyzing Model shows how to use InConst() as input for multiple other Analyzing Functions and how to provide the input of InConst() as parameter of an Analyzing Model.
Example with User Function	This Analyzing Model shows how to pass the value to InConst() when it is used as input interface of a User Function.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\InConst()")

Name	Description
Example for const	This Analyzing Script shows how to use the "const" statement as input for multiple Analyzing Functions.
Example with Analyzing Model	This Analyzing Script shows how to pass the value to InConst() when it is used as parameter of an Analyzing Model.

User Functions
(found in "..\User Function Files\Measurement Tasks\InConst()")

Name	Description
Example for InConst()	This User Function shows how to use InConst() as input for multiple other Analyzing Functions and how to provide the input of InConst() as input interface of a User Function.

7.4.18 Strings as Storage ID

7.4.18.1 General

This chapter shows how to use strings as storage ID within a Storage Profile. In this application, the string can be used as part of the file name and/or directory path of the to-be-stored offline data files.

7.4.18.2 Acquisition of Strings

Strings can be acquired from the following devices:

- ION SIMATIC S7 PN
- ION SIMATIC S7-1200
- ION SIMATIC S7-1500
- ION SIMATIC S7-300 T001
- ION SIMATIC S7-400 T001
- ION SIMATIC TDC T001
- ION SIMOTION T001

In addition, strings also can be generated within the Analyzing System or they can be received and forwarded by an Analyzing Function like InTCP().

7.4.18.3 Device Profiles

The following example shows a Device Profile for an ION SIMATIC S7 PN which contains one string definition:

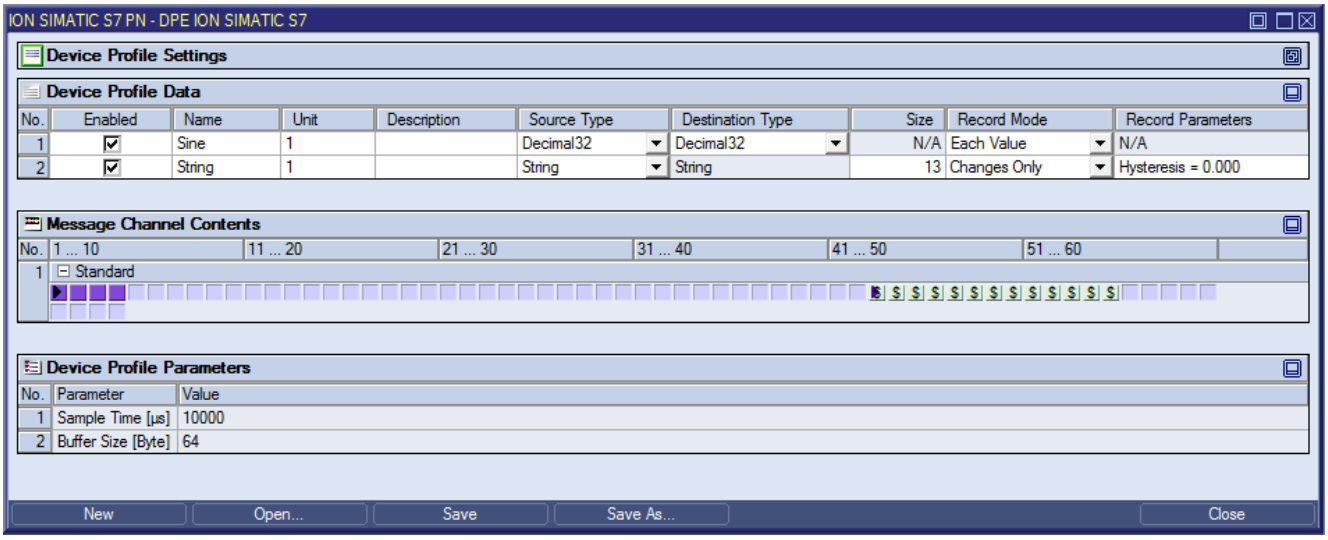

ION SIMATIC S7 PN - DPE ION SIMATIC S7
Device Profile Settings
Device Profile Data
No. Enabled Name Unit Description Source Type Destination Type Size Record Mode Record Parameters
1 Sine 1 Decimal32 Decimal32 N/A Each Value N/A
2 String 1 String String 13 Changes Only Hysteresis = 0.000
Message Channel Contents
No. 1 ... 10 11 ... 20 21 ... 30 31 ... 40 41 ... 50 51 ... 60
1 Standard
Device Profile Parameters
No. Parameter Value
1 Sample Time [µs] 10000
2 Buffer Size [Byte] 64
New
Open...
Save
Save As...
Close

Figure 148: The Device Profile "ION SIMATIC S7 PN" from the Example Location

Start the Interface Profile "..\Measurement Tasks\Strings as Storage ID\Simulation T001" and the Device Profile "..\Measurement Tasks\Strings as Storage ID\ION SIMATIC S7 PN" from the example location in order to get an online data of type string.

7.4.18.4 Analyzing Models

The following example shows an Analyzing Model which outputs the seconds of the day as Integer and as string:

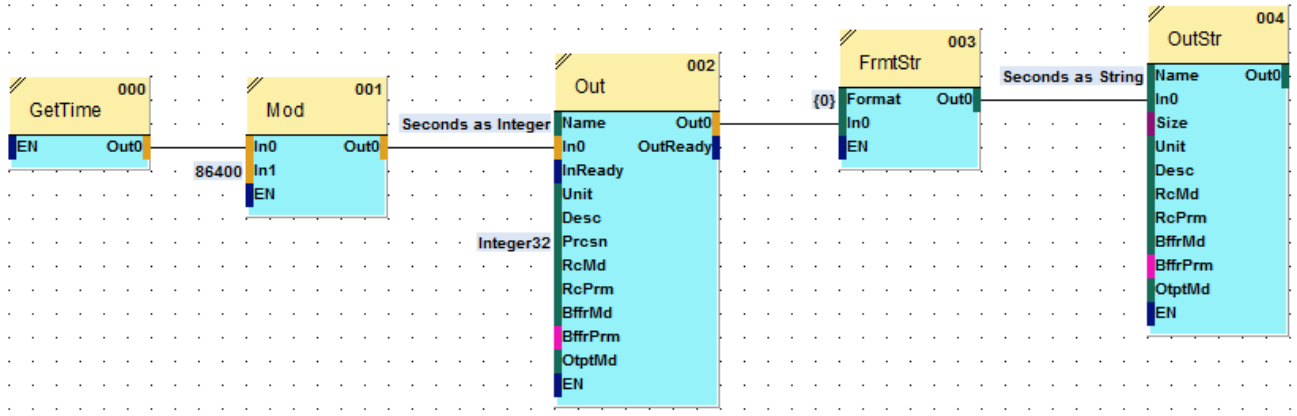


Figure 149: The Analyzing Model "Storage ID" from the Example Location

Start the Analyzing Model "..\Measurement Tasks\Strings as Storage ID\Storage ID" from the example location in order to get another online data of type string.

7.4.18.5 Storage Profiles

The following example shows a Storage Profile which uses the string data from the above Device Profile as "Storage ID 1" and the string data from the above Analyzing Model as "Storage ID 2":

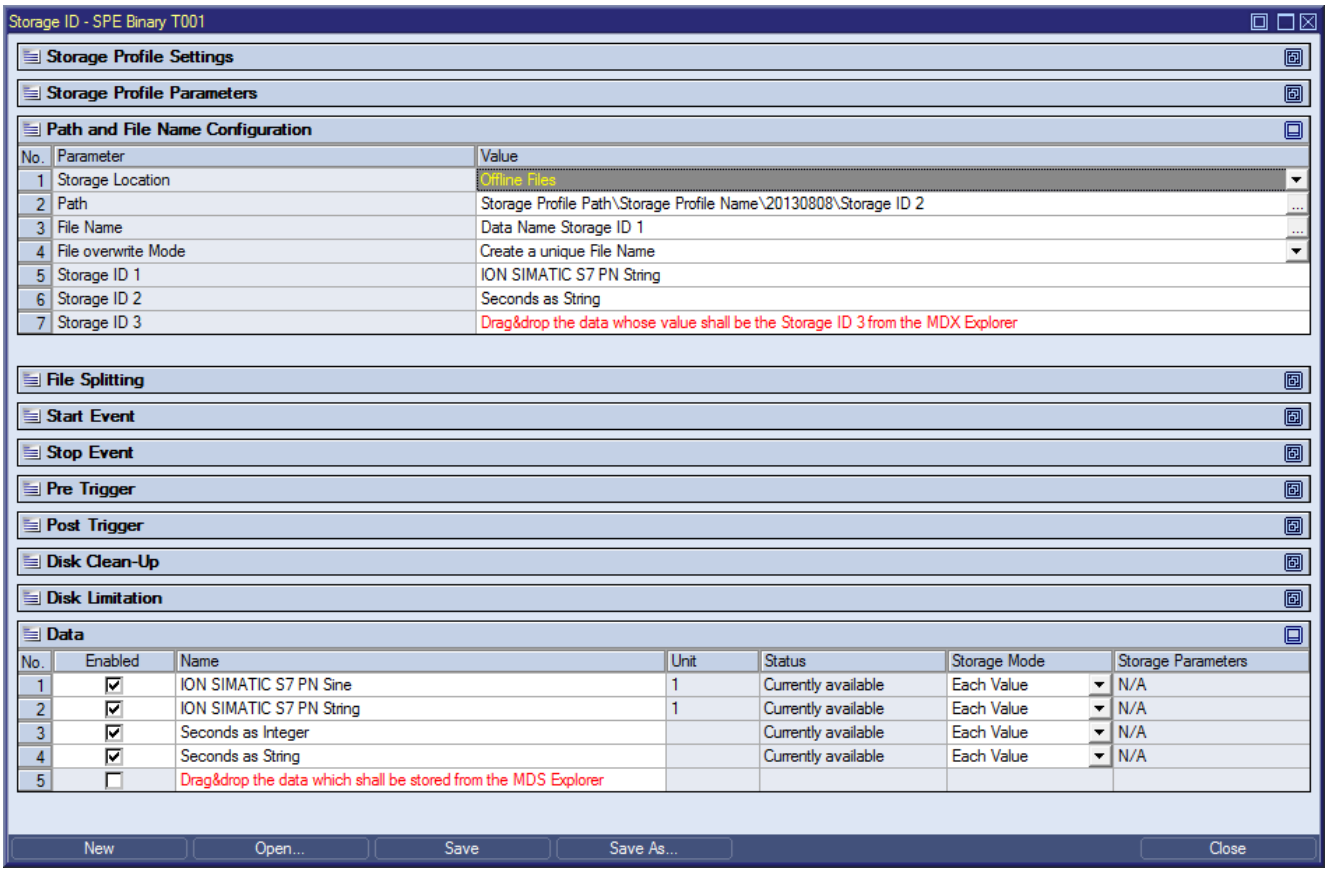

Storage ID - SPE Binary T001
Storage Profile Settings
Storage Profile Parameters
Path and File Name Configuration
No. Parameter Value
1 Storage Location Offline Files
2 Path Storage Profile Path\Storage Profile Name\20130808\Storage ID 2
3 File Name Data Name Storage ID 1
4 File overwrite Mode Create a unique File Name
5 Storage ID 1 ION SIMATIC S7 PN String
6 Storage ID 2 Seconds as String
7 Storage ID 3 Drag&drop the data whose value shall be the Storage ID 3 from the MDX Explorer
File Splitting
Start Event
Stop Event
Pre Trigger
Post Trigger
Disk Clean-Up
Disk Limitation
Data
No. Enabled Name Unit Status Storage Mode Storage Parameters
1 ION SIMATIC S7 PN Sine 1 Currently available Each Value N/A
2 ION SIMATIC S7 PN String 1 Currently available Each Value N/A
3 Seconds as Integer Currently available Each Value N/A
4 Seconds as String Currently available Each Value N/A
5 Drag&drop the data which shall be stored from the MDS Explorer
New
Open...
Save
Save As...
Close

Figure 150: The Storage Profile "Storage ID" from the Example Location

Start the Storage Profile "..\Measurement Tasks\Strings as Storage ID\Storage ID" from the example location and let it run for a few minutes.

7.4.18.6 Offline Data Files

The above Storage Profile creates offline data files like in the following screenshot:

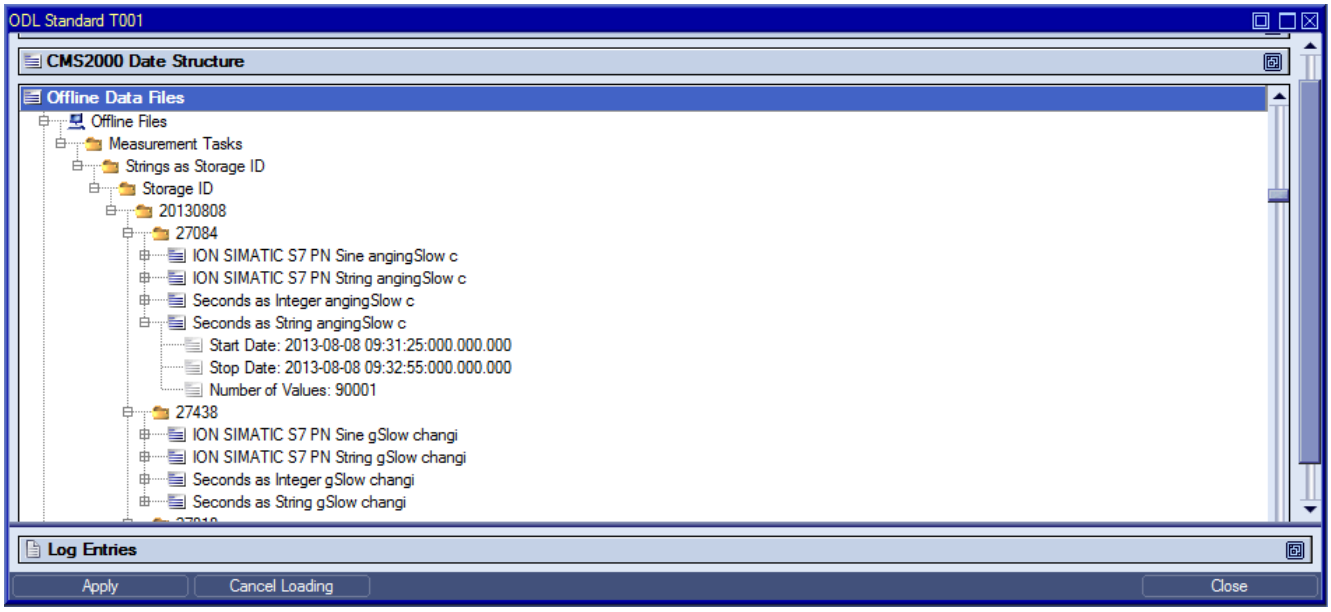


Figure 151: Example for the Offline Data Files which are created by the Storage Profile "Storage ID"

Each file name contains "Storage ID 1" (= the value of the data "ION SIMATIC S7 PN String") as additional part after the name of the actual data. In addition, the name of the last subdirectory is defined through "Storage ID 2" (= the value of the data "Seconds as String").

The values of the data which are configured as storage ID are taken from the moment which matches the start date of the offline data file.

7.4.18.7 Summary

The following Configuration Files are being used by this chapter:

Interface Profiles
(found in "..\Interface Profile Files\Measurement Tasks\Strings as Storage ID")

Name	Description
Simulation T001	This Interface Profile created a simulated ION SIMATIC S7 PN.

Device Profiles
(found in "..\Device Profile Files\Measurement Tasks\Strings as Storage ID")

Name	Description
ION SIMATIC S7 PN	This Device Profile creates two online data, one decimal and one string.

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Strings as Storage ID")

Name	Description
Storage ID	This Analyzing Model creats two online data, one Integer and one string.

Storage Profiles
(found in "..\Storage Profile Files\Measurement Tasks\Strings as Storage ID")

Name	Description
Storage ID	This Storage Profile uses the string data from the Device Profile and from the Analyzing Model as part of the file and directory names of the offline data files which it creates.

7.4.19 Performance Monitoring

7.4.19.1 General

This chapter shows how to use the built-in Analyzing Functions in order to run a basic performance monitoring for the ***X-Tools Client***, for the ***X-Tools Server*** and for the whole system.

7.4.19.2 Available Performance Data

The following Analyzing Functions are available for the acquisition of performance data:

- MonDisk() – MonitorDisk()
- MonSM() – MonitorSystemMemory()
- MonST() – MonitorSystemTimes()
- MonFrBlck() – MonitorFreeBlock()
- MonPC() – MonitorPerformanceCounter()
- MonPM() – MonitorProcessMemory()
- MonPT() – MonitorProcessTimes()
- IsPrcsBsy() – IsProcessBusy()

Refer to the documentation of each Analyzing Function for detailed information about their usage. Each Analyzing Function can be used within Analyzing Models and/or Analyzing Scripts in order to acquire the desired performance information.

7.4.19.3 Analyzing Scripts

X-Tools is delivered with example Analyzing Scripts which allow to start the acquisition of the performance information about the system, the ***X-Tools Server*** and the ***X-Tools Client***. These Analyzing Scripts are found below "..\Measurement Tasks\Performance Monitoring\Start Acquisition".

The following example shows an Analyzing Script which starts the acquisition of the performance information about the system:

```
var main()
{
    // start all Analyzing Models which are relevant for the acquisition of performance data about
    // the system
    Start( "://Measurement Tasks/Performance Monitoring/System/Disk C.amf.xts",  "DoNotWait" );
    Start( "://Measurement Tasks/Performance Monitoring/System/MonSM().amf.xts", "DoNotWait" );
    Start( "://Measurement Tasks/Performance Monitoring/System/MonST().amf.xts", "DoNotWait" );

    // start the Storage Profile which stores the performance data of the system
    Start( "://Measurement Tasks/Performance Monitoring/System.spf.xts", "DoNotWait" );

}
```

Figure 152: The Analyzing Script "Start Acquisition\System" from the Example Location

Similary, there are also Analyzing Scripts which allow to stop the acquisition of the performance information about the system, the ***X-Tools Server*** and the ***X-Tools Client***. These Analyzing Scripts are found below "..\Measurement Tasks\Performance Monitoring\Stop Acquisition".

The following example shows an Analyzing Script which stops the acquisition of the performance information about the system:

```
var main()
{
    // stop the Storage Profile which stores the performance data of the system
    Stop( "://Measurement Tasks/Performance Monitoring/System.spf.xts" );

    // start stop Analyzing Models which are relevant for the acquisition of performance data about
    // the system
    Stop( "://Measurement Tasks/Performance Monitoring/System/Disk C.amf.xts" );
    Stop( "://Measurement Tasks/Performance Monitoring/System/MonSM().amf.xts" );
```

```
    Stop( "://Measurement Tasks/Performance Monitoring/System/MonST().amf.xts" );

}
```

Figure 153: The Analyzing Script "Stop Acquisition\System" from the Example Location

There is always one Analyzing Script for the start/stop of the acquisition about the system, the ***X-Tools Server*** and the ***X-Tools Client***. Therefore, the performance monitoring can be started for each component separately or for all of them simultaneously.

The acquisition of the performance monitoring of all components can be started by calling of the context menu above the "Start Acquisition" directory and by choosing of "Start" afterwards. This starts all three Analyzing Scripts, which then start all of the necessary Analyzing Models and Storage Profiles.

The acquisition of the performance monitoring of all components can be stopped by calling of the context menu above the "Stop Acquisition" directory and by choosing of "Start" afterwards. This starts all three Analyzing Scripts, which then stop all of the running Storage Profiles and Analyzing Models.

7.4.19.4 Analyzing Models

The example from "..\Measurement Tasks\Performance Monitoring" uses Analyzing Models for the actual acquisition of the performance information. Generally, it is possible to create according Analyzing Scripts for exactly the same tasks.

Although it is very convenient to start all needed Analyzing Models and the matching Storage Profile via the prepared Analyzing Scripts, it of course is also possible to start each of the prepared Analyzing Models manually.

Each of the provided Analyzing Models uses one Analyzing Function for the acquisition of one type of performance information. For example, the following Analyzing Model determines the maximal free memory block which is available for the ***X-Tools Server***:

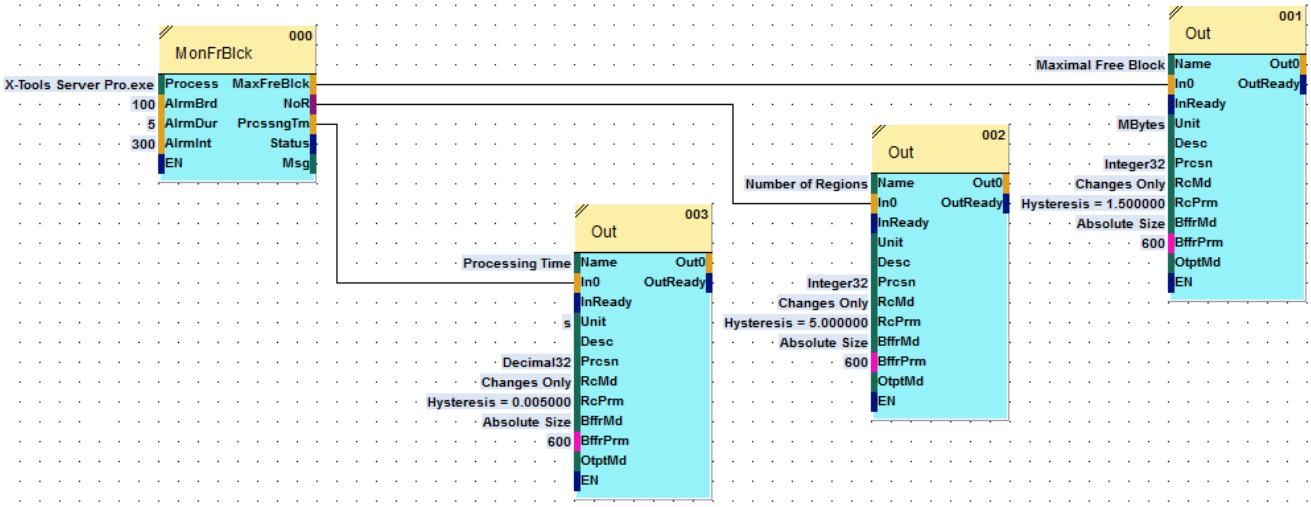


Figure 154: The Analyzing Model "X-Tools Server\MonFrBlck()" from the Example Location

In case not all of the provided Analyzing Models are needed for the current analysis, feel free to start only the needed Analyzing Models.

The provided Analyzing Models are designed for the performance monitoring of the system, the ***X-Tools Server*** and the ***X-Tools Client***. However, the process-dependent Analyzing Functions like MonFrBlck(), MonPC(), MonPM() and MonPT() also can be used for the monitoring of any other windows processes.

7.4.19.5 Storage Profiles

The Storage Profiles below "..\Measurement Tasks\Performance Monitoring" can be used in order to store all of the data which is acquired by the provided Analyzing Models.

7.4.19.6 Monitoring Views

The Monitoring Views below "..\Measurement Tasks\Performance Monitoring" can be used in order show a few of the acquired performance data:

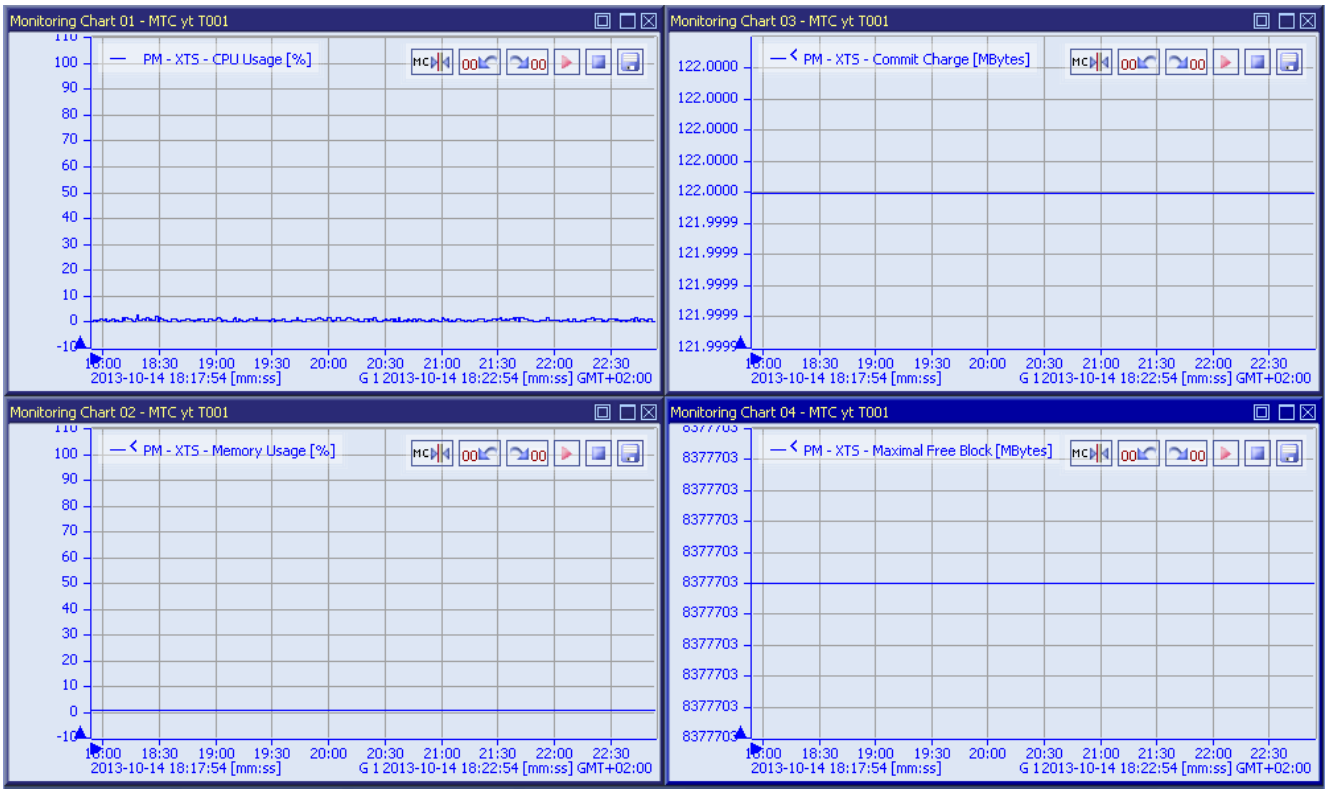

Monitoring Chart 01 - MTC yt T001
PM - XTS - CPU Usage [%]
Monitoring Chart 03 - MTC yt T001
PM - XTS - Commit Charge [MBytes]
Monitoring Chart 02 - MTC yt T001
PM - XTS - Memory Usage [%]
Monitoring Chart 04 - MTC yt T001
PM - XTS - Maximal Free Block [MBytes]
2013-10-14 18:17:54 [mm:ss]
G 1 2013-10-14 18:22:54 [mm:ss] GMT+02:00

Figure 155: Example output of the Monitoring View "X-Tools Server"

7.4.19.7 Summary

The following Configuration Files are being used by this chapter:

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Performance Monitoring")

Name	Description
System\Disk C	This Analyzing Model monitors the free space at disk C:.
System\MonSM()	This Analyzing Model monitors the memory usage through the system.
System\MonST()	This Analyzing Model monitors the CPU usage through the system.
X-Tools Client\IsPrcsBsy()	This Analyzing Model monitors whether the ***X-Tools Client*** is busy or not.
X-Tools Client\MonFrBlck()	This Analyzing Model monitors the maximal free memory block of the ***X-Tools Client***.
X-Tools Client\MonPC()	This Analyzing Model monitors the handle count and thread count of the ***X-Tools Client***.
X-Tools Client\MonPM()	This Analyzing Model monitors the memory usage through the ***X-Tools Client***.
X-Tools Client\MonPT()	This Analyzing Model monitors the CPU usage through the ***X-Tools Client***.
X-Tools Server\MonFrBlck()	This Analyzing Model monitors the maximal free memory block of the ***X-Tools Server***.
X-Tools Server\MonPC()	This Analyzing Model monitors the handle count and thread count of the ***X-Tools Server***.
X-Tools Server\MonPM()	This Analyzing Model monitors the memory usage through the ***X-Tools Server***.
X-Tools Server\MonPT()	This Analyzing Model monitors the CPU usage through the ***X-Tools Server***.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\Performance Monitoring")

Name	Description
Start Acquisition\System	This Analyzing Script starts the acquisition of the performance information about the system.
Start Acquisition\X-Tools Client	This Analyzing Script starts the acquisition of the performance information about the ***X-Tools Client***.
Start Acquisition\X-Tools Server	This Analyzing Script starts the acquisition of the performance information about the ***X-Tools Server***.
Stop Acquisition\System	This Analyzing Script stops the acquisition of the performance information about the system.
Stop Acquisition\X-Tools Client	This Analyzing Script stops the acquisition of the performance information about the ***X-Tools Client***.
Stop Acquisition\X-Tools Server	This Analyzing Script stops the acquisition of the performance information about the ***X-Tools Server***.

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\Performance Monitoring")

Name	Description
System	This Monitoring View shows some performance information about the system.
X-Tools Client	This Monitoring View shows some performance information about the ***X-Tools Client***.
X-Tools Server	This Monitoring View shows some performance information about the ***X-Tools Server***.

Storage Profiles
(found in "..\Storage Profile Files\Measurement Tasks\Performance Monitoring")

Name	Description
System	This Storage Profile stores the performance information about the system.
X-Tools Client	This Storage Profile stores the performance information about the ***X-Tools Client***.
X-Tools Server	This Storage Profile stores the performance information about the ***X-Tools Server***.

7.4.20 Editing of Scalar Data.csv

7.4.20.1 General

This chapter shows how to edit the permant storage of scalar data outside of ***X-Tools*** with Microsoft Excel. The format of the “Scalar Data.csv” file which is used for the permanent storage of scalar data through ***X-Tools*** must provide a certain format in order to be readable through ***X-Tools***. The following chapters describe the rules which must be considered when this file is modified outside of ***X-Tools***.

7.4.20.2 Import to Microsoft Excel

Create a few scalar data inside ***X-Tools*** and terminate ***X-Tools*** afterwards. The Analyzing Script “Generate scalar Data” from the “..\Measurement Tasks\Editing of Scalar Data.csv” directory can be used for this task, but any other scalar data can be used as well.

After the termination of ***X-Tools***, the file “Scalar Data.csv” can be found below the “..\Server\User\Scalar Data” directory of your ***X-Tools*** installation.

Microsoft Excel 2010 has been used for the following screenshots, but any other version of Excel can be used as well. It is also possible to use an application different to Microsoft Excel, but the detailed import and export settings must be determined separately then.

Perform the following steps for the import:

- Start Microsoft Excel.
- Open a new, blank workbook.
- Go to “Data” -> “Get external Data” and choose “From Text”.
- Choose the “Scalar Data.csv” file from the “..\Server\User\Scalar Data” directory of your ***X-Tools*** installation as input file.
- At the first step of the text import wizard, choose “Delimited” as file type and press “Next”:

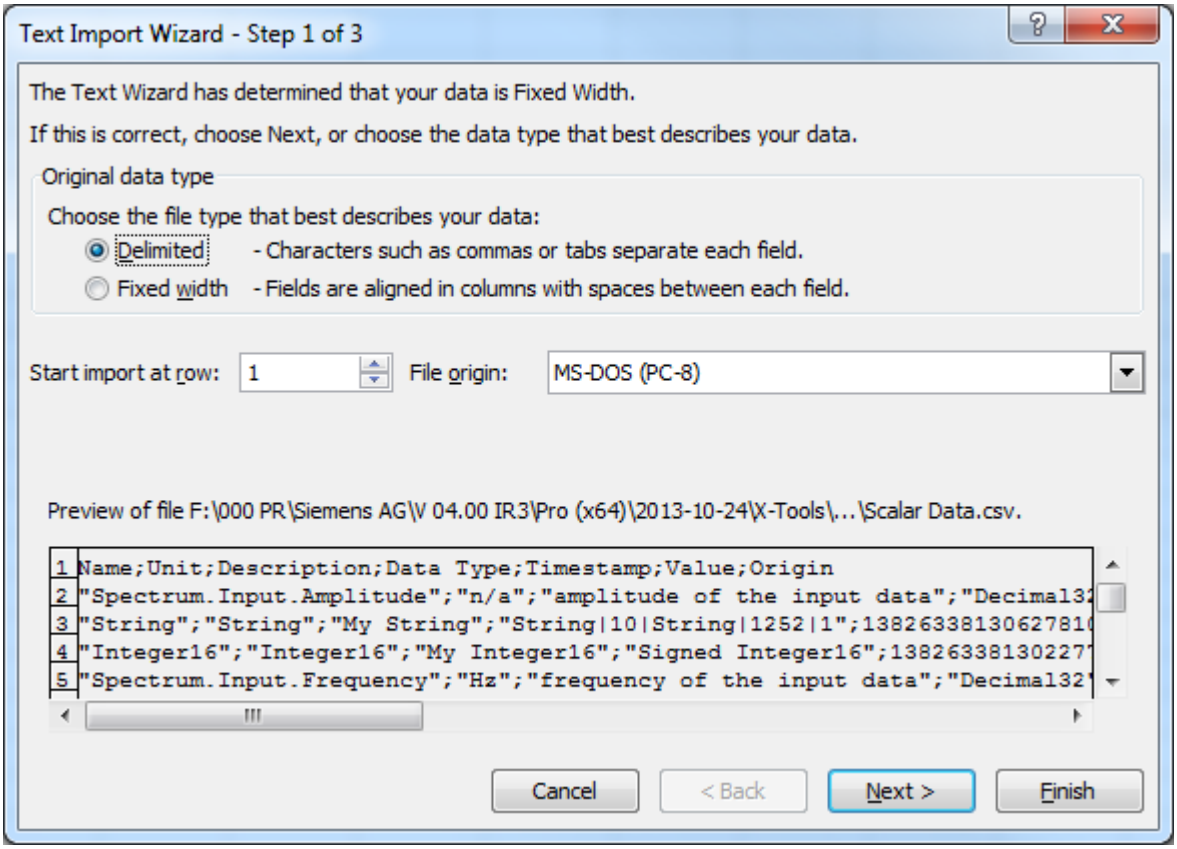

Text Import Wizard - Step 1 of 3
The Text Wizard has determined that your data is Fixed Width.
If this is correct, choose Next, or choose the data type that best describes your data.
Original data type
Choose the file type that best describes your data:
Delimited - Characters such as commas or tabs separate each field.
Fixed width - Fields are aligned in columns with spaces between each field.
Start import at row: 1
File origin: MS-DOS (PC-8)
Preview of file F:\000 PR\Siemens AG\V 04.00 IR3\Pro (x64)\2013-10-24\X-Tools\...\Scalar Data.csv.
1 Name;Unit;Description;Data Type;Timestamp;Value;Origin
2 "Spectrum.Input.Amplitude";"n/a";"amplitude of the input data";"Decimal32
3 "String";"String";"My String";"String|10|String|1252|1";13826338130627810
4 "Integer16";"Integer16";"My Integer16";"Signed Integer16";13826338130227
5 "Spectrum.Input.Frequency";"Hz";"frequency of the input data";"Decimal32
Cancel
< Back
Next >
Finish

Figure 156: Import via Microsoft Excel - Step 1

- At the second step of the text import wizard, choose only “Semicolon” as delimiter and press “Next”:

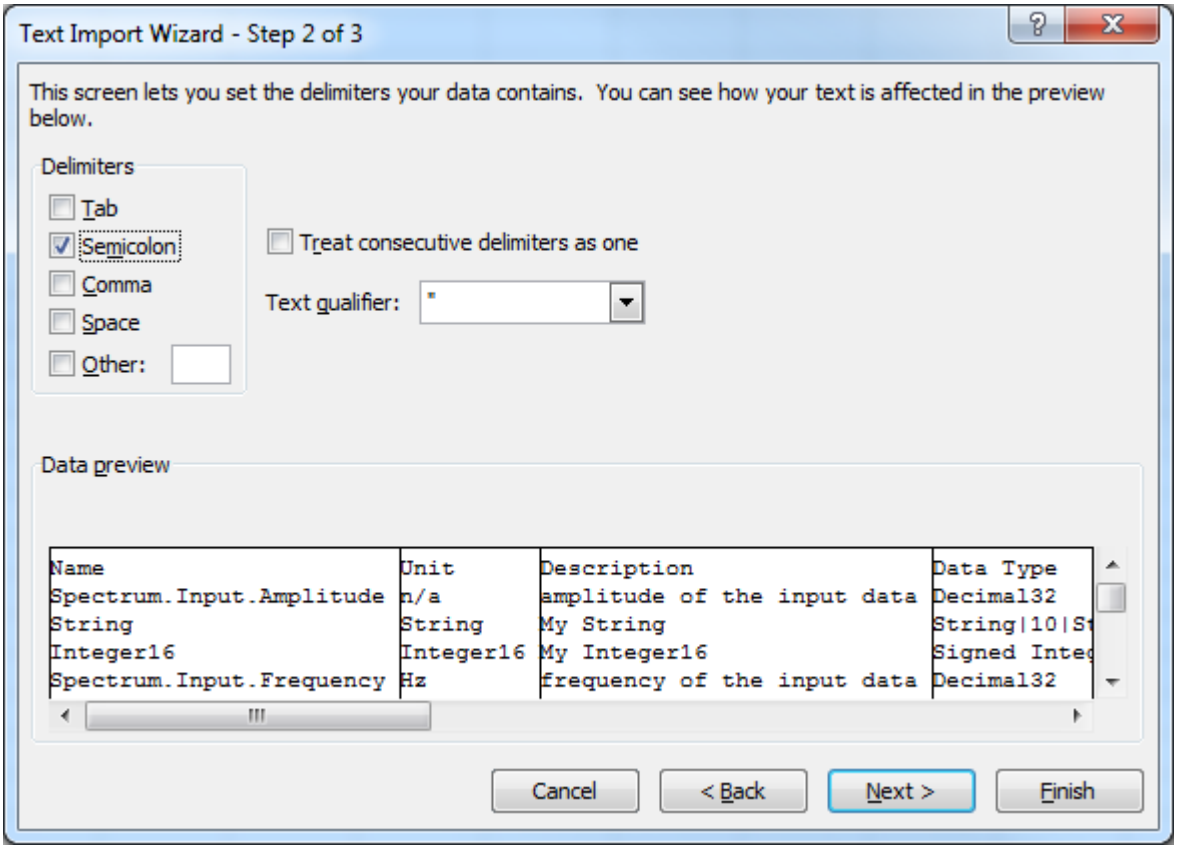

Text Import Wizard - Step 2 of 3
This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.
Delimiters
Tab
Semicolon
Comma
Space
Other:
Treat consecutive delimiters as one
Text qualifier:
Data preview
Name Unit Description Data Type
Spectrum.Input.Amplitude n/a amplitude of the input data Decimal32
String String My String String|10|S
Integer16 Integer16 My Integer16 Signed Inte
Spectrum.Input.Frequency Hz frequency of the input data Decimal32
Cancel
< Back
Next >
Finish

Figure 157: Import via Microsoft Excel - Step 2

- At the third step of the text import wizard, select all columns within the “Data preview” group, choose “Text” as “Column data format” afterwards and press “Finish” then:

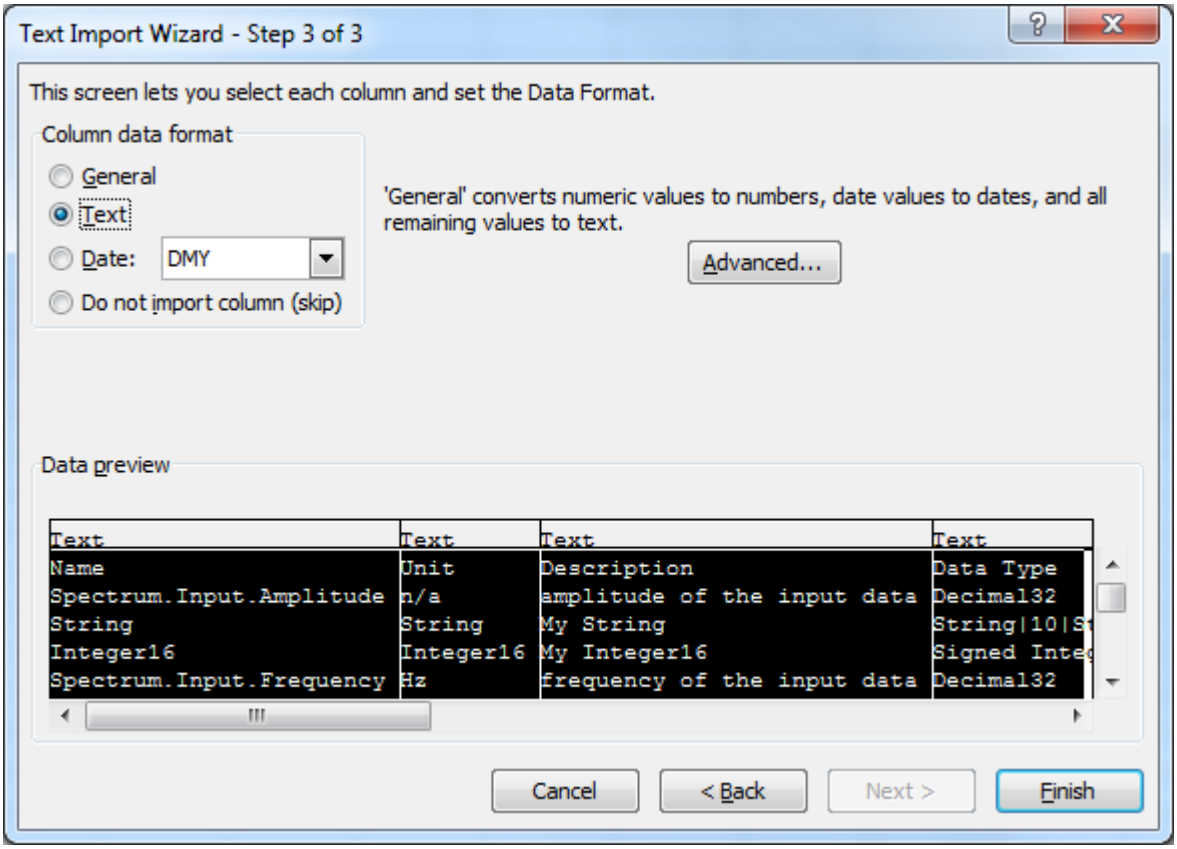

Text Import Wizard - Step 3 of 3
This screen lets you select each column and set the Data Format.
Column data format
General
Text
Date: DMY
Do not import column (skip)
'General' converts numeric values to numbers, date values to dates, and all remaining values to text.
Advanced...
Data preview
Text Text Text Text
Name Unit Description Data Type
Spectrum.Input.Amplitude n/a amplitude of the input data Decimal32
String String My String String|10|S
Integer16 Integer16 My Integer16 Signed Inte
Spectrum.Input.Frequency Hz frequency of the input data Decimal32
Cancel
< Back
Next >
Finish

Figure 158: Import via Microsoft Excel - Step 3

- Choose to put the to-be-imported data into the position “A1” of the existing, blank worksheet and press “OK”:

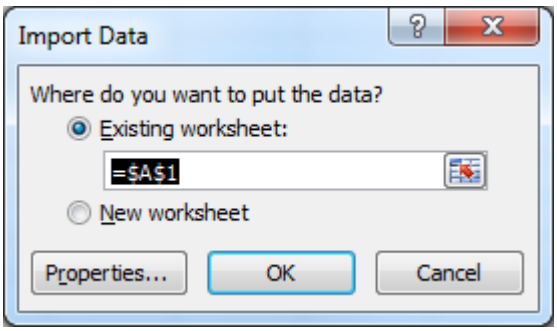


Figure 159: Import via Microsoft Excel - Step 4

- After the above steps, the contents of “Scalar Data.csv” have been imported to Microsoft Excel and look like in the following example:

	A	B	C	D	E	F	G	H
1	Name	Unit	Description	Data Type	Timestamp	Value	Origin	
2	Spectrum.Input.Amplitude	n/a	amplitude of the input data	Decimal32	1382633813068282100	10	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
3	String	String	My String	String|10|String|1252|1	1382633813062781000	Xr 10	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
4	Integer16	Integer16	My Integer16	Signed Integer16	1382633813022773000	10	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
5	Spectrum.Input.Frequency	Hz	frequency of the input data	Decimal32	1382633813073783200	5	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
6	Integer32	Integer32	My Integer32	Signed Integer32	1382633813028274100	10	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
7	Boolean	Boolean	My Boolean	Boolean	1382633813010770600	1	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
8	Spectrum.Output.Unit	n/a	unit of the output data	String|64|String|1252|1	1382633813091786800	My Unit	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
9	Spectrum.Output.Description	n/a	description of the output data	String|64|String|1252|1	1382633813097788000	My Description	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
10	Decimal32	Decimal32	My Decimal32	Decimal32	1382633813039276300	10	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
11	Spectrum.Output.Name	n/a	name of the output data	String|16|String|1252|1	1382633813085785600	My Spectrum	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
12	Integer08	Integer08	My Integer08	Signed Integer8	1382633813016771800	10	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
13	Spectrum.Calculation.NoP	NoP	number of points of the calculation	Signed Integer16	1382633813080284500	2048	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
14	Decimal64	Decimal64	My Decimal64	Decimal64	1382633813045277500	10	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	
15	Integer64	Integer64	My Integer64	Signed Integer64	1382633813033275100	10	Example Location (trunk)://Measurement Tasks/Editing of Scalar Data.csv/Generate scalar Data.asf.xts	

Figure 160: Import via Microsoft Excel - Step 5

It is possible now to change the name, unit, description, data type, timestamp, value and origin of each present scalar data as well as to add and to remove rows with scalar data information.

In case any other way for the import of “Scalar Data.csv” is used it must be ensured that Excel (or another used application) does not modify the contained timestamp and values.

7.4.20.3 Export from Microsoft Excel

In order to export the modified information about scalar data from Microsoft Excel, the following steps must be performed:

- Go to “File” -> “Save As”.
- Choose “CSV (MS-DOS) (*.csv)” as file type, specify some file name and press “Save”.

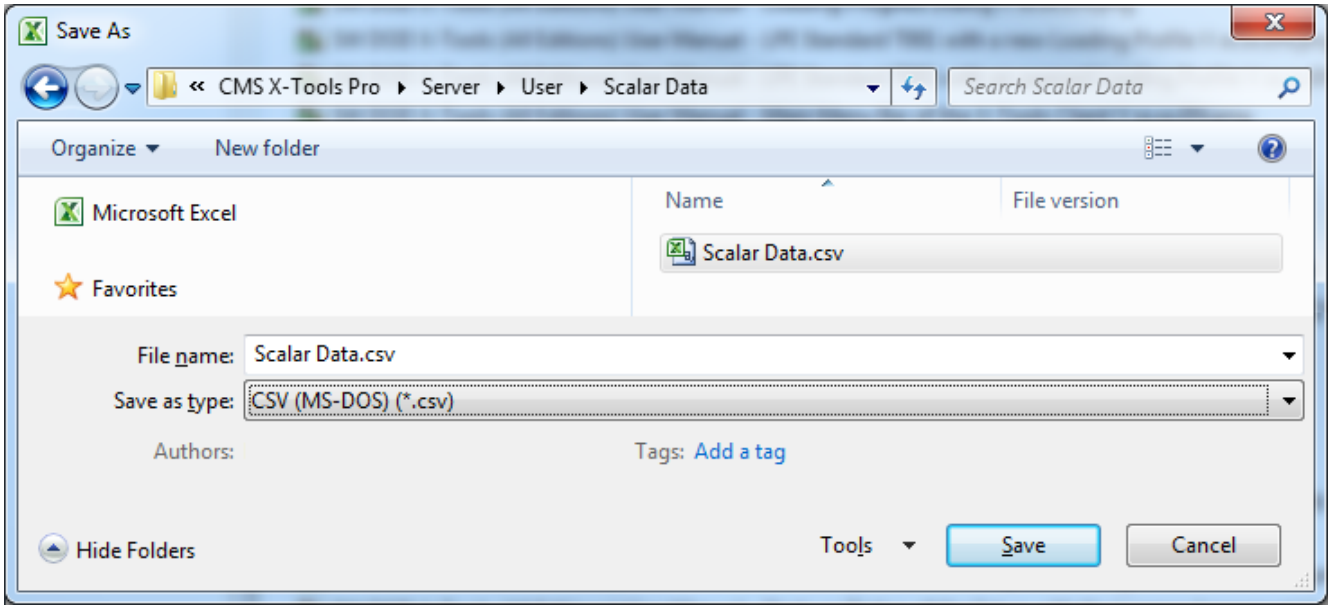


Figure 161: Export via Microsoft Excel - Step 1

- Press “OK” at the upcoming message box in order to tell Excel that it is OK for you that the target file will not contain multiple sheets.

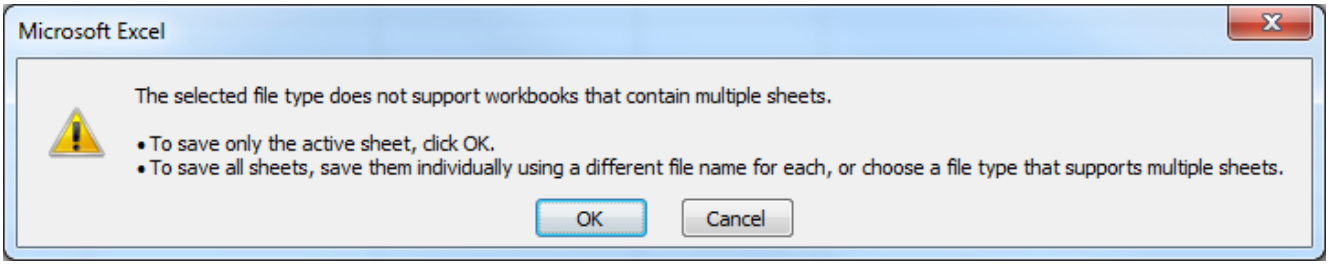

Microsoft Excel
The selected file type does not support workbooks that contain multiple sheets.
• To save only the active sheet, click OK.
• To save all sheets, save them individually using a different file name for each, or choose a file type that supports multiple sheets.
OK
Cancel

Figure 162: Export via Microsoft Excel - Step 2

- Press “Yes” at the next message box in order to tell Excel that it is OK for you that the target file will not contain anything which is not compatible to CSV.

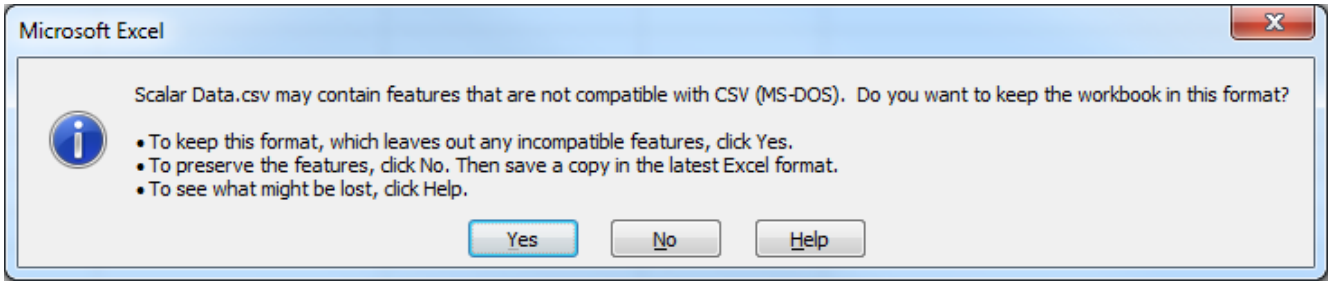

Microsoft Excel
Scalar Data.csv may contain features that are not compatible with CSV (MS-DOS). Do you want to keep the workbook in this format?
• To keep this format, which leaves out any incompatible features, click Yes.
• To preserve the features, click No. Then save a copy in the latest Excel format.
• To see what might be lost, click Help.
Yes
No
Help

Figure 163: Export via Microsoft Excel - Step 3

After these steps, Excel has saved the edited scalar data as .csv file again. You can copy the Excel-generated file to the “..\Server\User\Scalar Data” directory of your ***X-Tools*** installation with a name of “Scalar Data.csv” now. Afterwards, ***X-Tools*** can be started and reads the to-be-used scalar data from the update file.

In case errors within this file are detected through ***X-Tools***, according log entries are written to the system log and a backup of the erroneous “Scalar Data.csv” file is created by ***X-Tools***.

7.4.21 Massive sequential processing with Scalar Data

7.4.21.1 General

Some applications have to perform always the same processing for different parts (= measurement points) of a plant or process. In such a scenario, the ability of ***X-Tools*** to pass parameters from Analyzing Scripts to other Configuration Files (e.g. to Analyzing Models) can be used in order to use always the identical Configuration Files for the processing of different data with or without different parameters.

In addition, the MTC Mimic Board T001 can be used in order to view and to edit Scalar Data. These Scalar Data can hold the parameters for the later processing of each measurement point. In case of very large applications, hundreds or even thousands of parameters may be needed. In order to avoid to maintain the same number of Scalar Data, this chapter also shows how the currently configured parameters can be stored from Scalar Data into a file and how parameters can be loaded again from this file.

The Analyzing Models, Analyzing Scripts and Monitoring Views to which the following chapters refer are found within the "..\Measurement Tasks\Massive sequential processing with Scalar Data" folders of the example location which is delivered together with ***X-Tools***.

7.4.21.2 Preparation of an Example File

The following example shows an Analyzing Script which prepares a file with paramaters for the processing of three different measurement points:

```
#include "://Measurement Tasks\\Massive sequential processing with Scalar Data\\00 Helpers.asf.xts"


var main()
{
    // save some default values for measurement point 1
    WrtVlToFl(  File, 1, Frequency1,     10 );
    WrtVlToFl(  File, 1, Frequency2,     20 );
    WrtVlToFl(  File, 1, Frequency3,     25 );
    WrtStrToFl( File, 1, VibrationName,  "Vibration 1" );
    WrtVlToFl(  File, 1, CycleTime,      1 );
    WrtVlToFl(  File, 1, NumberOfPoints, 2048 );
    WrtStrToFl( File, 1, SpectrumName,   "Spectrum 1" );

    // save some default values for measurement point 2
    WrtVlToFl(  File, 2, Frequency1,     50 );
    WrtVlToFl(  File, 2, Frequency2,     75 );
    WrtVlToFl(  File, 2, Frequency3,     100 );
    WrtStrToFl( File, 2, VibrationName,  "Vibration 2" );
    WrtVlToFl(  File, 2, CycleTime,      1 );
    WrtVlToFl(  File, 2, NumberOfPoints, 4096 );
    WrtStrToFl( File, 2, SpectrumName,   "Spectrum 2" );

    // save some default values for measurement point 3
    WrtVlToFl(  File, 3, Frequency1,     20 );
    WrtVlToFl(  File, 3, Frequency2,     30 );
    WrtVlToFl(  File, 3, Frequency3,     40 );
    WrtStrToFl( File, 3, VibrationName,  "Vibration 3" );
    WrtVlToFl(  File, 3, CycleTime,      0.5 );
    WrtVlToFl(  File, 3, NumberOfPoints, 2048 );
    WrtStrToFl( File, 3, SpectrumName,   "Spectrum 3" );

}
```

Figure 164: The Analyzing Script "10 Prepare File" from the Example Location

The Analyzing Script “00 Helpers” contains basic definitions about the path to the to-be-used file as well as the names of the to-be-stored and to-be-read values. In addition, it also provides functions which store the current values of the used Scalar Data to the file and which read the current values from the file and write them as new values of the used Scalar Data:

```
const    File           = "C:\\Temp\\Default.ini";
const    Frequency1     = "Frequency1";
const    Frequency2     = "Frequency2";
const    Frequency3     = "Frequency3";
const    VibrationName  = "VibrationName";
const    CycleTime      = "CycleTime";
const    NumberOfPoints = "NumberOfPoints";
const    SpectrumName   = "SpectrumName";


var LoadConfiguration()
{
    var    Key;


    Key = s'Measurement Point.Active Measurement Point';

    OutSclr(    "Generation.Frequency 1",       RdVlFrmFl(  File, Key, Frequency1 ),     "Hz" );
    OutSclr(    "Generation.Frequency 2",       RdVlFrmFl(  File, Key, Frequency2 ),     "Hz" );
    OutSclr(    "Generation.Frequency 3",       RdVlFrmFl(  File, Key, Frequency3 ),     "Hz" );
    OutSclrStr( "Generation.Vibration Name",    RdStrFrmFl( File, Key, VibrationName ),  256 );
    OutSclr(    "Calculation.Cycle Time",       RdVlFrmFl(  File, Key, CycleTime ),      "ms" );
    OutSclr(    "Calculation.Number of Points", RdVlFrmFl(  File, Key, NumberOfPoints ), "",  "",
                "Integer32" );
    OutSclrStr( "Calculation.Spectrum Name",    RdStrFrmFl( File, Key, SpectrumName ),   256 );
    return 0;

}


var SaveConfiguration()
{
    var    Key;


    Key = s'Measurement Point.Active Measurement Point';

    WrtVlToFl(  File, Key, Frequency1,     s'Generation.Frequency 1' );
    WrtVlToFl(  File, Key, Frequency2,     s'Generation.Frequency 2' );
    WrtVlToFl(  File, Key, Frequency3,     s'Generation.Frequency 3' );
    WrtStrToFl( File, Key, VibrationName,  s'Generation.Vibration Name' );
    WrtVlToFl(  File, Key, CycleTime,      s'Calculation.Cycle Time' );
    WrtVlToFl(  File, Key, NumberOfPoints, s'Calculation.Number of Points' );
    WrtStrToFl( File, Key, SpectrumName,   s'Calculation.Spectrum Name' );
    return 0;

}
```

Figure 165: The Analyzing Script “00 Helpers” from the Example Location

The Analyzing Functions WrtVlToFl() and WrtStrToFl() are used in order to write values and strings to the file. Accordingly, the Analyzing Functions RdVlFrmFl() and RdStrFrmFl() are used in order to read values and strings from the file.

Start the Analyzing Script “10 Prepare File” in order to create a default file at “C:\Temp\Default.ini”:

```
[1]
Frequency1=10.000000
Frequency2=20.000000
Frequency3=25.000000
VibrationName=Vibration 1
CycleTime=1.000000
NumberOfPoints=2048.000000
SpectrumName=Spectrum 1
Name=-2.611022

[2]
Frequency1=50.000000
Frequency2=75.000000
Frequency3=100.000000
VibrationName=Vibration 2
CycleTime=1.000000
NumberOfPoints=4096.000000
SpectrumName=Spectrum 2

[3]
Frequency1=20.000000
Frequency2=30.000000
Frequency3=40.000000
VibrationName=Vibration 3
CycleTime=0.500000
NumberOfPoints=2048.000000
SpectrumName=Spectrum 3
```

Figure 166: Contents of “C:\Temp\Default.ini” after running of the Analyzing Script “10 Prepare File”

This file is a pure text file and also can be edited outside of ***X-Tools*** with standard text editors. The format inside the file is identical to the format of standard .ini files of Windows.

7.4.21.3 Preparation of the Scalar Data

The following example shows an Analyzing Script which creates the Scalar Data that are used in order to view and to edit the parameters of each measurement point:

```
var main()
{
    // selection of the current measurement point
    OutSclr( "Measurement Point.Active Measurement Point", 1, "", "", "Integer32" );
    OutSclr( "Measurement Point.Save Configuration",       0, "", "", "Boolean" );
    OutSclr( "Measurement Point.Load Configuration",       0, "", "", "Boolean" );

    // generation of the input data
    OutSclr( "Generation.Frequency 1",        10, "Hz" );
    OutSclr( "Generation.Frequency 2",        20, "Hz" );
    OutSclr( "Generation.Frequency 3",        25, "Hz" );
    OutSclrStr( "Generation.Vibration Name", "Vibration 1", 256 );

    // calculation of the spectrum
    OutSclr( "Calculation.Cycle Time",          1, "ms" );
    OutSclr( "Calculation.Number of Points", 2048, "",  "", "Integer32" );
    OutSclrStr( "Calculation.Spectrum Name", "Spectrum 1", 256 );

}
```

Figure 167: The Analyzing Script “10 Prepare Scalar Data” from the Example Location

Start the Analyzing Script “10 Prepare Scalar Data” in order to add the used Scalar Data to the **MDS Explorer**.

7.4.21.4 Viewing, editing, saving and loading of the Parameters

Open the Monitoring View “Configure Measurement Points”:

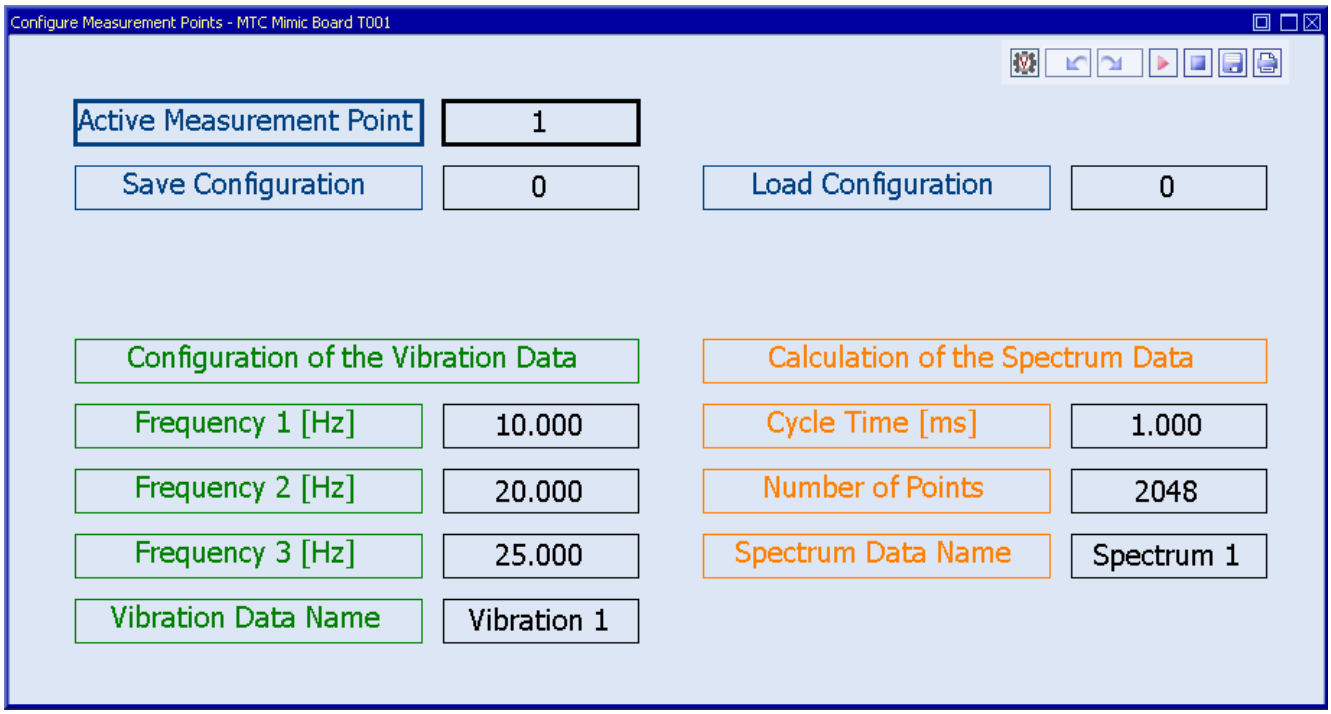


Figure 168: The Monitoring View “Configure Measurement Points” from the Example Location

This Monitoring View provides the following functionality:

- the current values of the Scalar Data can be viewed
  - the blue boxes contain the Scalar Data for the management of the parameters
  - the green boxes contain the parameters for the configuration of the vibration data of the currently active measurement point
  - the orange boxes contain the parameters for the calculation of the spectrum data of the currently active measurement point
- the current values of the Scalar Data can be edited
  - this is done via the green and orange boxes
- the currently active measurement point can be switched
  - this is done via the value right to “Active Measurement Point”
- the current values of the Scalar Data can be saved to file as parameters for the active measurement point
  - in order to save the currently configured values as parameters for the currently active measurement point, change the value right to “Save Configuration” from 0 to 1
  - the actual saving is performed by the Analyzing Script “20 Configure Measurement Points” – therefore, this Analyzing Script must be running when saving to file shall be available
  - after the Analyzing Script has saved the parameters to the file, it switches the value of “Save Configuration” back to 0
- the values from the file for the active measurement point can be loaded and written to the Scalar Data
  - in order to load the values for the currently active measurement point from the file and to write them into the Scalar Data, change the value right to “Load Configuration” from 0 to 1
  - the actual loading is performed by the Analyzing Script “20 Configure Measurement Points” – therefore, this Analyzing Script must be running when loading from file shall be available

- after the Analyzing Script has loaded the parameters from the file, it switches the value of “Load Configuration” back to 0
  - in addition, the values within the green and orange boxes are updated and show the values which were loaded from the file

```
#include "://Measurement Tasks\\Massive sequential processing with Scalar Data\\00 Helpers.asf.xts"


var main()
{
    while( 1 )
    {
        if( s'Measurement Point.Save Configuration' )
        {
            // we have to save the current configuration of the active measurement point
            SaveConfiguration();

            // show that we finished saving
            AddLE( "Measurement Point " + s'Measurement Point.Active Measurement Point' + " saved." );
            OutSclr( "Measurement Point.Save Configuration", 0, "", "", "Boolean" );
        }

        if( s'Measurement Point.Load Configuration' )
        {
            // we have to load the configuration of the active measurement point
            LoadConfiguration();

            // show that we finished loading
            AddLE( "Measurement Point " + s'Measurement Point.Active Measurement Point' + " loaded." );
            OutSclr( "Measurement Point.Load Configuration", 0, "", "", "Boolean" );
        }

        Sleep( 1 );
    }

}
```

Figure 169: The Analyzing Script “20 Configure Measurement Points” from the Example Location

In a typical scenario, the configuration of multiple measurement points could look like this:

- create the Scalar Data which are needed for the to-be-configured parameters of each measurement point
  - this can be done with an Analyzing Script like “10 Prepare Scalar Data”
- prepare a MTC Mimic Board T001 which views the to-be-configured parameters and which allows to edit them
  - an example for such a Monitoring View is “Configure Measurement Points”
- enter the desired parameters for a measurement point, set the number of the measurement point and save these parameters to the file
- continue to enter the desired parameters for the next measurement point, set the number of the new measurement point and save these parameters to the file
  - this can be done until the parameters of all measurement points have been configured

7.4.21.5 Run sequential processing via Scalar Data

After the to-be-used parameters for all measurement points have been set up and stored to a file, these parameters can be used for a very automatic, sequential processing of a virtually endless number of identical measurement points. The following Analyzing Script runs the processiong of three measurement points and uses Scalar Data in order to hold the parameters of each measurement point:

```
#include "://Measurement Tasks\\Massive sequential processing with Scalar Data\\00 Helpers.asf.xts"


var main()
{
    var    i;


    while( 1 )
    {
        for( i=1; i<=3; i++ )
        {
            // load the configuration for measurement point i
            OutSclr( "Measurement Point.Active Measurement Point", i, "", "", "Integer32" );
            LoadConfiguration();

            // start the processing
            StartEx( "://Measurement Tasks\\Massive sequential processing with Scalar Data\\
                      10 Generate Input Data.amf.xts", "DoNotWait", 0,
                     s'Generation.Vibration Name',
                     s'Generation.Frequency 1',
                     s'Generation.Frequency 2',
                     s'Generation.Frequency 3';);
            StartEx( "://Measurement Tasks\\Massive sequential processing with Scalar Data\\
                      20 Calculate Spectrum.amf.xts", "DoNotWait", 0,
                     s'Generation.Vibration Name',
                     s'Calculation.Number of Points',
                     s'Calculation.Spectrum Name',
                     s'Calculation.Cycle Time';);

            // let the calculation run for 10 seconds
            Sleep( 10 );

            // stop the processing
            Stop( "://Measurement Tasks\\Massive sequential processing with Scalar Data\\
                   20 Calculate Spectrum.amf.xts" );
            Stop( "://Measurement Tasks\\Massive sequential processing with Scalar Data\\
                   10 Generate Input Data.amf.xts" );
        }
    }

}
```

Figure 170: The Analyzing Script “50 Run sequential processing via Scalar Data” from the Example Location

Start this Analyzing Script in order to run the sequential processing. The Monitoring View “Configure Measurement Points” can be used in order to see always the currently active measurement point and its used parameters. In addition, the Monitoring View “Show Sequential processing” can be opened in order to see the vibration and spectrum data which are created by each measurement point.

The following screenshot shows an example of the output of the Monitoring View “Configure Measurement Points” when the Analyzing Script “50 Run sequential processing via Scalar Data” is running:

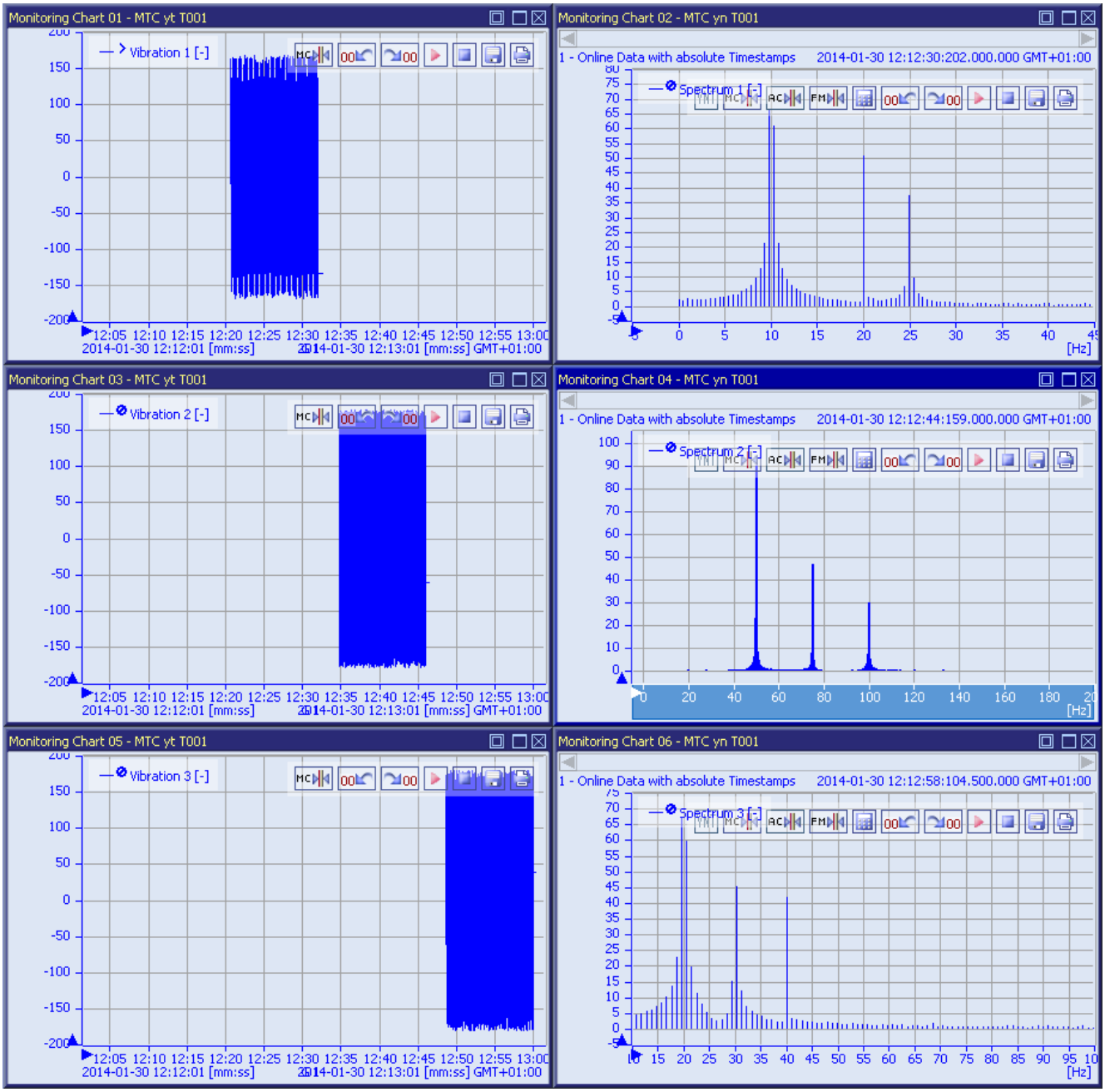

Monitoring Chart 01 - MTC yt T001
Vibration 1 [-]
Monitoring Chart 02 - MTC yn T001
1 - Online Data with absolute Timestamps
2014-01-30 12:12:30:202.000.000 GMT+01:00
Spectrum 1 [-]
[Hz]
Monitoring Chart 03 - MTC yt T001
Vibration 2 [-]
Monitoring Chart 04 - MTC yn T001
1 - Online Data with absolute Timestamps
2014-01-30 12:12:44:159.000.000 GMT+01:00
Spectrum 2 [-]
[Hz]
Monitoring Chart 05 - MTC yt T001
Vibration 3 [-]
Monitoring Chart 06 - MTC yn T001
1 - Online Data with absolute Timestamps
2014-01-30 12:12:58:104.500.000 GMT+01:00
Spectrum 3 [-]
[Hz]
2014-01-30 12:12:01 [mm:ss]
2014-01-30 12:13:01 [mm:ss] GMT+01:00

Figure 171: The Analyzing Script “50 Run sequential processing via Scalar Data” from the Example Location

Stop the Analyzing Script “50 Run sequential processing via Scalar Data” and all Analyzing Models before you continue to the next chapter.

7.4.21.6 Run sequential processing via local Variables

Instead of using of Scalar Data for holding of the parameters for the currently to-be-processed measurement point, also local variables of the Analyzing Script can be used. Whether Scalar Data or local variables are more suited for a certain application can be dependent to application and/or to other requirements and has to be judged for each case separatelly.

The following Analyzing Script runs the processiong of three measurement points and uses local variables in order to hold the parameters of each measurement point. The output of the called Analyzing Models is identical to the example with Scalar Data from point 7.4.21.5:

```
#include "://Measurement Tasks\\Massive sequential processing with Scalar Data\\00 Helpers.asf.xts"


var main()
{
    var    i;
    var    tmpFrequency1;
    var    tmpFrequency2;
    var    tmpFrequency3;
    var    tmpVibrationName;
    var    tmpCycleTime;
    var    tmpNumberOfPoints;
    var    tmpSpectrumName;


    while( 1 )
    {
        for( i=1; i<=3; i++ )
        {
            // load the configuration for measurement point i
            tmpFrequency1     = RdVlFrmFl(  File, i, Frequency1 );
            tmpFrequency2     = RdVlFrmFl(  File, i, Frequency2 );
            tmpFrequency3     = RdVlFrmFl(  File, i, Frequency3 );
            tmpVibrationName  = RdStrFrmFl( File, i, VibrationName );
            tmpCycleTime      = RdVlFrmFl(  File, i, CycleTime );
            tmpNumberOfPoints = RdVlFrmFl(  File, i, NumberOfPoints );
            tmpSpectrumName   = RdStrFrmFl( File, i, SpectrumName );

            // start the processing
            StartEx( "://Measurement Tasks\\Massive sequential processing with Scalar Data\\
                      10 Generate Input Data.amf.xts", "DoNotWait", 0,
                     tmpVibrationName,
                     tmpFrequency1,
                     tmpFrequency2,
                     tmpFrequency3;);
            StartEx( "://Measurement Tasks\\Massive sequential processing with Scalar Data\\
                      20 Calculate Spectrum.amf.xts", "DoNotWait", 0,
                     tmpVibrationName,
                     tmpNumberOfPoints,
                     tmpSpectrumName,
                     tmpCycleTime;);

            // let the calculation run for 10 seconds
            Sleep( 10 );

            // stop the processing
            Stop( "://Measurement Tasks\\Massive sequential processing with Scalar Data\\
                   20 Calculate Spectrum.amf.xts" );
            Stop( "://Measurement Tasks\\Massive sequential processing with Scalar Data\\
                   10 Generate Input Data.amf.xts" );
        }
    }

}
```

Figure 172: The Analyzing Script “50 Run sequential processing via local Variables” from the Example Location

Start this Analyzing Script in order to run the sequential processing. As this Analyzing Script does not set the values of any Scalar Data, the displayed values within the Monitoring View “Configure Measurement Points” are never updated and do not provide information about the currently processed measurement point and the currently used parameters. The Monitoring View “Show Sequential processing” can be opened in order to see the vibration and spectrum data which are created by each measurement point and its displayed data curves are identical to point 7.4.21.5.

7.4.21.7 Additional Notes

The above examples show a rather simple case of a tiny measurement point. However, the described functionality can be extended in order to cover more kinds of Configuration Files (Device Profiles, Storage Profiles, …) and/or more parameters. It also is possible to replace the used Integer number of the measurement point through some string or ID.

The created parameters can not be used only for sequential processing but also for parallel processing of multiple measurement points – in this case, probably the variant of the Analyzing Script with local variables is more suitable.

7.4.21.8 Summary

The following Configuration Files are being used by this chapter:

Analyzing Models
(found in "..\Analyzing Model Files\Measurement Tasks\Massive sequential processing with Scalar Data")

Name	Description
10 Generate Input Data	This Analyzing Model receives parameters from the Analyzing Script which starts it. The output of this Analyzing Model is used as input for the calculation of the spectrum.
20 Calculate Spectrum	This Analyzing Model receives parameters from the Analyzing Script which starts it. The Analyzing Model calculates the frequency spectrum for the given input data.

Analyzing Scripts
(found in "..\Analyzing Script Files\Measurement Tasks\ Massive sequential processing with Scalar Data")

Name	Description
00 Helpers	This Analyzing Script provides definitions and functions which are used by multiple other Analyzing Scripts.
10 Prepare File	This Analyzing Script prepares a file which contains the parameters for three measurement points.
10 Prepare Scalar Data	This Analyzing Script adds the Scalar Data which are used for the viewing and editing of the parameters of one measurement point.
20 Configure Measurement Points	This Analyzing Script does the actual saving and loading of parameters from Scalar Data to the file and from the file to Scalar Data.
50 Run sequential processing via local Variables	This Analyzing Script runs a sequential processing of three measurement tasks and uses local variables for holding of the parameters of the currently active measurement point.
50 Run sequential processing via Scalar Data	This Analyzing Script runs a sequential processing of three measurement tasks and uses Scalar Data for holding of the parameters of the currently active measurement point.

Monitoring Views
(found in "..\Monitoring View Files\Measurement Tasks\ Massive sequential processing with Scalar Data")

Name	Description
Configure Measurement Points	This Monitoring View allows to view and to edit the parameters of one measurement point.
Show sequential processing	This Monitoring View shows the output from the sequential processing of all three measurement points.

8 Reference Part - Overview

8.1 X-Tools Client

8.1.1 Controls

8.1.1.1 Tables

Tables are being used mainly for editing and configuration purposes. The tables of the ***X-Tools Client*** provide the following main functionalities:

- Context Menu
- Keyboard Operations
- Sizeable Columns
- Shiftable Columns
- Minimize and Maximize

Context Menu

The context menu of each table can be called via a click onto the title bar with the right mouse button. The following context menu items are provided:

Context Menu Item	Description
Restore Column Defaults	restores the default configuration of all columns as it is after the installation of ***X-Tools***
AutoFit Column Widths	automatically fits the width of all columns so that all contained texts are visible
Show/Hide > …	allows to show or hide the columns of the table; the provided sub-menu contains all of the columns of the table which can be shown or hidden by the user
Autosize Table	optimizes the width of all visible columns
Minimize	minimizes the table
Maximize	maximizes the table

Keyboard Operations

Different keyboard operations are supported depending to whether the table is in editing mode at the moment or not. The following keyboard operations are provided in case the table is not in editing mode:

Keyboard Operation	Description
<Del>	performs a remove/delete operation
<Ctrl> + <C>	performs a copy operation
<Ctrl> + <X>	performs a cut operation
<Ctrl> + <V>	performs a paste operation
<Enter>	puts the cell which owns the input focus currently into editing mode and highlights its current value
<Backspace>	puts the cell which owns the input focus currently into editing mode and clears its current value
<Cursor left>	moves the input focus one column to the left
<Cursor right>	moves the input focus one column to the right
<Cursor up>	moves the input focus one row to the top
<Cursor down>	moves the input focus one row to the bottom
<Tab>	jumps from the current table to the next table/tree/menu bar
<Shift> + <Tab>	jumps from the current table to the previous table/tree/menu bar

The following keyboard operations are provided in case the table is in editing mode:

Keyboard Operation	Description
<Del>	performs a remove operation
<Ctrl> + <C>	performs a copy operation
<Ctrl> + <X>	performs a cut operation
<Ctrl> + <V>	performs a paste operation
<Esc>	discards the changed value and puts the cell which owns the input focus currently into not-editing mode
<Enter>	takes over the changed value and puts the cell which owns the input focus currently into not-editing mode
<Backspace>	puts the cell which owns the input focus currently into editing mode and clears its current value
<Cursor left>	moves the input focus left to the next editable cell of the same row
<Cursor right>	moves the input focus right to the next editable cell of the same row
<Cursor up>	moves the input focus up to the next editable cell of the same column
<Cursor down>	moves the input focus down to the next editable cell of the same column
<Tab>	jumps from the current cell to the next editable cell at the right side
<Shift> + <Tab>	jumps from the current cell to the next editable cell at the left side

Sizeable Columns

The user is able to change the width of each column through a left mouse click in between the title cells of two columns. Through moving of the mouse, the width of the left column can be changed as long as the left mouse button is kept pressed.

Shiftable Columns

The user is able to change the position of columns among each other through a left mouse click into the title cell of a column. Trough moving the mouse into between the title cells of two other columns, the dragged column can be put to its new position. In case the dragged column is released outside of its parent table, the dragged column is hidden.

Minimize and Maximize

In order to minimize or maximize a table, a double-click with the left mouse button onto its title bar can be performed. Additionally, the icon at the right border of the title bar of each table also can be used in order to minimize and maximize the table.

8.1.1.2 Trees

Trees are being used mainly for visualization and selection purposes. The trees of the ***X-Tools Client*** provide the following main functionalities:

- Keyboard Operations

Keyboard Operations

Different keyboard operations are supported depending to whether the tree is in editing mode at the moment or not. The following keyboard operations are provided in case the tree is not in editing mode:

Keyboard Operation	Description
<Del>	performs a remove/delete operation
<Ctrl> + <C>	performs a copy operation
<Ctrl> + <X>	performs a cut operation
<Ctrl> + <V>	performs a paste operation
<Esc>	cancels the current selection
<F2>	puts the item which owns the input focus currently into editing mode
<Tab>	jumps from the current tree to the next tree/table/menu bar
<Shift> + <Tab>	jumps from the current tree to the previous tree/table/menu bar
<Cursor left> onto an expanded tree item	collapses the tree item
<Cursor left> onto a collapsed tree item	moves to the parent item of the tree item
<Cursor right> onto a collapsed tree item	expands the tree item
<Cursor right> onto an expanded tree item	moves the input focus to the first item below the expanded tree item
<Cursor up>	moves the input focus to the next upper tree item without performing of any collapsing or expanding
<Cursor down>	moves the input focus to the next lower tree item without performing of any collapsing or expanding
<Backspace>	moves the input focus to the parent item of the current tree item without performing of any collapsing or expanding
<Home>	moves the input focus to the topmost tree item
<End>	moves the input focus to the bottommost tree item

The following keyboard operations are provided in case the tree is in editing mode:

Keyboard Operation	Description
<Del>	performs a remove operation
<Ctrl> + <C>	performs a copy operation
<Ctrl> + <X>	performs a cut operation
<Ctrl> + <V>	performs a paste operation
<Esc>	discards the changed value and puts the cell which owns the input focus currently into not-editing mode
<Enter>	takes over the changed value and puts the cell which owns the input focus currently into not-editing mode
<Tab>	behaves like <Enter>

8.1.1.3 Menu Bars

Menu bars are found at the bottom of most dialogs and provide the following main functionalities:

- Keyboard Operations

Keyboard Operations

The following keyboard operations are provided by each menu bar:

Keyboard Operation	Description
<Esc>	behaves like if the “Yes” or “OK” button would have been clicked
<Enter>	behaves like if the “No” or “Cancel” button would have been clicked
<Cursor left>	moves the input focus from the current button to the next button at the left
<Cursor right>	moves the input focus from the current button to the next button at the right
<Cursor up>	moves the input focus from the current button to the next button at the left
<Cursor down>	moves the input focus from the current button to the next button at the right
<Tab>	moves the input focus from the current button to the next button at the right; in case the rightmost button had the input focus already, the input focus is being moved to the topmost control of the dialog
<Shift> + <Tab>	moves the input focus from the current button to the next button at the left; in case the leftmost button had the input focus already, the input focus is being moved to the bottommost control of the dialog

8.2 X-Tools Server

8.2.1 Overview

The ***X-Tools Server*** does not provide its own user interface. Instead, only an icon is displayed within the task bar of Windows. This icon provides a tool tip and a context menu, which can be called in order to receive the most basic information about the ***X-Tools Server*** and to configure the most basic settings.

All further configurations are being performed via the ***X-Tools Client***. See point 6.3 for detailed information about how to connect one or multiple ***X-Tools Clients*** to the ***X-Tools Server***.

8.2.2 Icon of the X-Tools Server

The icon of the ***X-Tools Server*** provides basic information about the current status of the ***X-Tools Server***. The following status information can be provided via this icon:

Icon	Description
(startup)	The ***X-Tools Server*** is starting up.
(normal operation)	The ***X-Tools Server*** is running normally or shutting down.
(safe mode)	The ***X-Tools Server*** is running in safe mode.
(error)	The ***X-Tools Server*** has detected a severe error during its startup.

Safe Mode

Detailed information about the safe mode of the ***X-Tools Server*** is found within the tutorial (see point 6.19).

Error

In case a severe error has been detected during the startup, more detailed information is provided via the tool tip of the icon and via the **X-Tools Server Status** dialog (see point 8.2.5).

8.2.3 Tool Tip

In order to show the tool tip of the ***X-Tools Server***, simply place the mouse cursor above the icon of the ***X-Tools Server***. After a few moments, the tool tip is being displayed automatically.

The tool tip provides basic information like the version of the ***X-Tools Server*** and its current status.

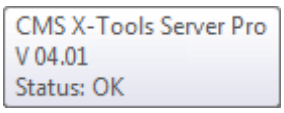


*Figure 173: Example of a Tool Tip of the **X-Tools Server***

8.2.4 Context Menu

In order to call the context menu of the ***X-Tools Server***, click onto the icon of the ***X-Tools Server*** with the right mouse button. The following context menu items are provided:

Context Menu Item	Description
Show X-Tools Server Status…	opens the **X-Tools Server Status** dialog (see point 8.2.5)
Configure X-Tools Server Connectivity…	opens the **X-Tools Server Connectivity** dialog (see point 6.3.6)
Re-initialize the X-Tools Server	re-initializes the ***X-Tools Server***, which terminates all modules and restarts them afterwards
Exit	terminates the ***X-Tools Server***

8.2.5 X-Tools Server Status Dialog

The **X-Tools Server Status** dialog can be called from the context menu of the ***X-Tools Server***, which is available from the task bar of Windows. This dialog displays the current status of the ***X-Tools Server*** as well as detailed information about the version, edition and platform.

In case warnings or errors have been detected during the startup of the ***X-Tools Server***, this information also is being displayed by the **X-Tools Server Status** dialog. As the X-Tools Server does not provide any other user interface, the **X-Tools Server Status** dialog can be used in order to receive a short overview about the ongoing operations without the need to connect an ***X-Tools Client***.

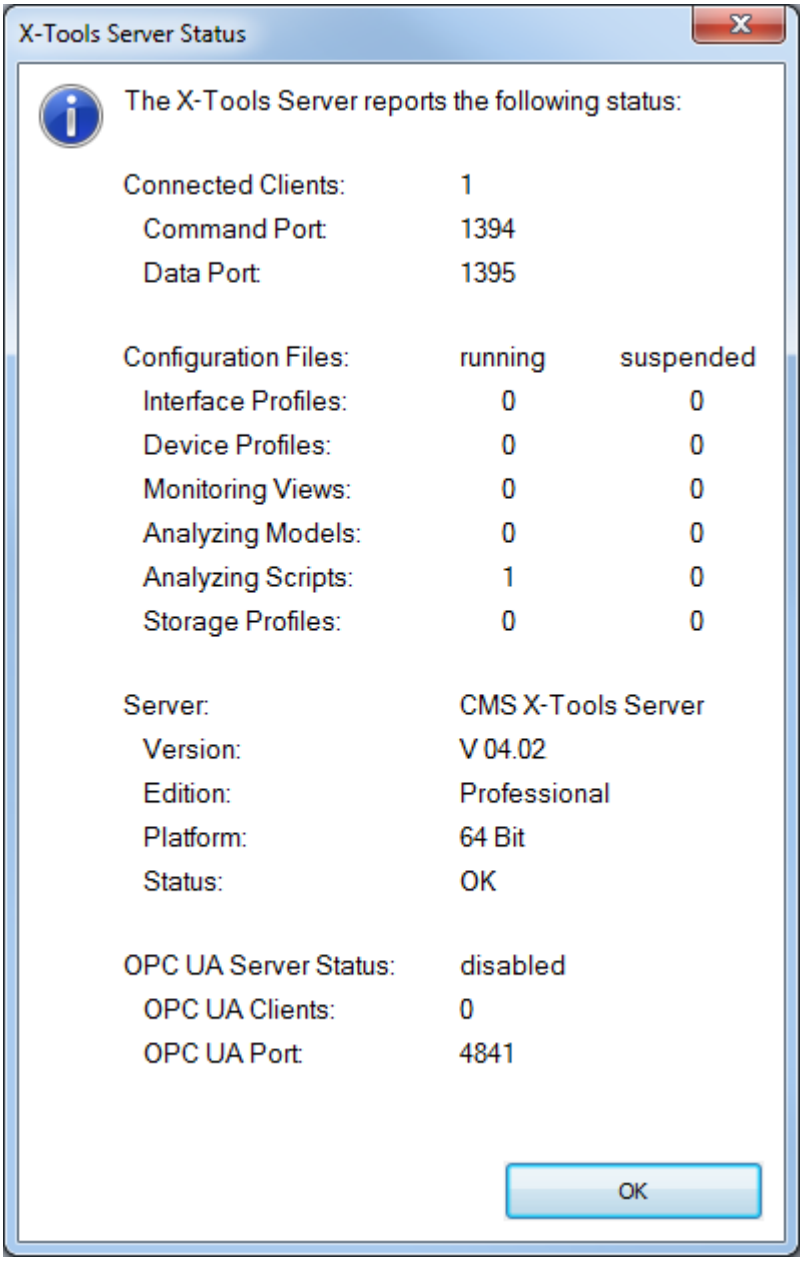


*Figure 174: **X-Tools Server Status** Dialog*

8.3 X-Tools Tray

8.3.1 Overview

The ***X-Tools Tray*** application manages the tray icon of the ***X-Tools Server***. In addition, it provides the context menu of the tray icon and the **X-Tools Server Status** and **X-Tools Server Connectivity** dialogs.

The ***X-Tools Tray*** is started automatically by different mechanisms and keeps running also in case the rest of ***X-Tools*** does not run at the moment. In case the ***X-Tools Server*** is not present, the task tray icon of the ***X-Tools Tray*** is hidden.

At operating systems which allow multiple Windows session to be present simultaneously (e.g. Microsoft Windows Server 2003, but with restrictions also Microsoft Windows XP and Microsoft Windows 7), one instance of the ***X-Tools Tray*** can be started in each Windows session. This approach allows to monitor and control the one-and-only ***X-Tools Server*** from any of the present Windows sessions.

The ***X-Tools Tray*** application also can be started manually. However, typically this is not needed and it is not possible to run multiple instances of the ***X-Tools Tray*** within one Windows session simultaneously.

8.4 X-Tools Service

8.4.1 Overview

The ***X-Tools Service*** can be used for the following tasks:

- automatic start of the ***X-Tools Server*** during the startup of Windows
- monitoring of the ***X-Tools Server*** during its runtime

8.4.2 Automatic Start of the X-Tools Server during the Startup of Windows

The ***X-Tools Service*** can start the ***X-Tools Server*** during the startup of Windows. In order to enable this functionality, the Automated Startup option from the global options of the ***X-Tools Server*** must be configured to “Start the X-Tools Server before a user logs in to Windows”.

Enabling of this option installs the ***X-Tools Service***, which becomes visible within the **Services** dialog of Windows then.

8.4.3 Monitoring of the X-Tools Server during its Runtime

The ***X-Tools Service*** can monitor whether the ***X-Tools Server*** is running properly. In order to enable this functionality, the Automated Restart option from the global options of the ***X-Tools Server*** must be configured to “Automatically restart the X-Tools Server in case it stops to operate normally”.

Enabling of this option installs the ***X-Tools Service***, which becomes visible within the **Services** dialog of Windows then.

While the monitoring is active, the ***X-Tools Service*** constantly checks whether the ***X-Tools Server*** is still responsive. In case the ***X-Tools Server*** does not respond for a too long time, the ***X-Tools Service*** kills the process of the ***X-Tools Server*** and starts a new instance of the ***X-Tools Server***. After such an event, the ***X-Tools Server*** will start up in safe mode.

8.4.4 Stopping of the X-Tools Service

The ***X-Tools Service*** is being stopped by the ***X-Tools Service*** in case none of the above functionalities is needed. The ***X-Tools Service*** also is being stopped in case the ***X-Tools Service*** is being shut down manually by the user.

While the ***X-Tools Server*** is running and in case the ***X-Tools Service*** is enabled via the global options, the ***X-Tools Server*** also monitors the ***X-Tools Service*** and attempts to start it whenever it does not run - therefore, the ***X-Tools Service*** always must be stopped via the global options of the ***X-Tools Server***, a manual attempt to stop the ***X-Tools Service*** will not work in most cases.

9 List of Abbreviations

Abbreviation	Description
AME	Analyzing Model Editor
ANF	Analyzing Function
ANS	Analyzing System
ANX	Analyzing System Explorer
ASE	Analyzing Script Editor
CMM	Communication Module
CTN	Control Node
DMS	Device Management System
DMX	Device Management System Explorer
DPE	Device Profile Editor
GOE	Global Options Editor
IFM	Interface Module
IFN	Interface Node
IOM	I/O-Module, Input/Output-Module
ION	I/O-Node, Input/Output-Node
IPE	Interface Profile Editor
LPE	Loading Profile Editor
MDS	Master Data System
MDX	Master Data System Explorer
MMS	Main Management System
MMX	Main Management System Explorer
MPE	Main Profile Editor
MTC	Monitoring Chart
MTS	Monitoring System
MTX	Monitoring System Explorer
MVE	Monitoring View Editor
ODL	Offline Data Loader
UAE	User Accounts Editor
SLV	System Log Viewer
SPE	Storage Profile Editor
STS	Storage System
STX	Storage System Explorer
UFE	User Function Editor
UPS	Uninterruptible Power Supply
USF	User Function
XTC	X-Tools Client
XTS	X-Tools Server
XTT	X-Tools Tray
XTV	X-Tools Service

10 Contact Information

Should you have any questions concerning the software application, please refer to the Industry Sector Technical Support.

Department

Siemens AG

Industry Sector

Phone

+49 (0) 911 895 7222 (Monday to Friday, 09:00 am to 05:00 pm CET/CEST)

Fax

+49 (0) 911 895 7223

Internet

www.siemens.de/automation/service&support

Thank you for using one of the above mentioned contacts to ensure your inquiry is registered and can be processed.